

المؤسسة الجامعية للدراسات والنشر والتوزيع - جدة

جلد ریغیر

# البرمجة : لغة المؤول ( الاسمبلر )

ترجمة د. عبد الحسن الحسيني



**البرمجة بلغة المؤول**  
**( الاسمبار )**

جميع الحقوق محفوظة  
الطبعة الأولى  
1410 هـ - 1990 م

 المكتبة الوطنية والارشيف  
المملكة العربية السعودية

بيروت - الجمهورية العربية السورية - شارع الفوق - ص.ب. ١١١١

هاتف: ١٠٠٠٠٠٠ - ١٠٠٠٠٠٠ - ١٠٠٠٠٠٠

بيروت - الجمهورية العربية السورية - شارع الفوق - ص.ب. ١١١١

هاتف: ١٠٠٠٠٠٠ - ١٠٠٠٠٠٠ - ١٠٠٠٠٠٠

سلسلة بإشراف  
د. عبد الحسن الحسيني

جاك ريفيير

# البرمجة : لغة المؤول ( الاسمبار )

ترجمة د. عبد الحسن الحسيني

منشور  
الجامعة اللبنانية - بيروت

هذا الكتاب ترجمة :

**LA PROGRAMMATION  
EN ASSEMBLEUR**

Par

**Jacques RIVIERE**

## تقديم

تعتبر لغة أسيمبلر (المؤول) من اللغات الفعالة وذات الإمكانيات الكبيرة نظراً لأنها تسمح للمبرمج باستعمال جميع إمكانيات ومقدرات وموارد الحاسب، كما تسمح له بالدخول إلى «قلب» الآلة والعمل بالمرافق الداخلية للحاسب، مما يضيف على البرنامج المكتوب بهذه اللغة فعالية كبيرة خصوصاً فيما يتعلق بالدقة والسرعة والعمل في الوقت الفعلي (real time) المستعمل كثيراً لإدارة العمليات الصناعية.

هذا الكتاب يُعالج لغة أسيمبلر الخاصة بعائلة الحاسبات IBM 360/370 التي شهدت إنتشاراً واسعاً في حقل المعلوماتية وأحدثت ثورة في صناعة الحاسبات في السنوات الأخيرة وبقيت تركيبة وهيكلية هذه الآلات مُستعملة وصالحة في وقتنا هذا وجرى إستعمالها والإفادة منها حتى في صناعة المعالج الصغري وتصميم الميكروحاسبات.

وبالنسبة للمبرجة بلغة المؤول، فإن تقنية هذه البرمجة لا تختلف أبداً من آلة إلى أخرى، صغيرة كانت أم كبيرة، معالجاً صغرياً أو نظاماً كبيراً. أما الفرق الوحيد فيكون في كون كود - الآلة يختلف من آلة إلى أخرى، أما طريقة العمل والمعالجة وإستعمال المرافق والذاكرة فلا تختلف إلا في عدد المرافق المبلوغة من المبرمج، وبالتالي فإن التصرف على أي مؤول يبقى صالحاً بالنسبة لمعالج آخر مؤول آخر.

وهنا يجب الإشارة إلى أن مؤول IBM/370 يتألف من أكبر عدد ممكن من التعليمات، وعدد مرافق الحاسب يعادل 16 للمعطيات و16 للعناوين ويستعمل عدداً كبيراً من طرق العنوان، يصلح قسم منها لعنونة المعلومات عند إستعمال المعالج الصغري.

المترجم



## تمهيد

لماذا كتاب جديد يختص بلغة المؤول (Assembler) ؟ وما هو المؤول ؟ هل تعرفون مبرمجين يعملون بلغة المؤول حتى الآن ، بينما تقدم اللغات المتطورة إمكانيات وتسهيلات جديدة ؟

كثيراً ما نسمع جميع هذه الأسئلة إضافة إلى أخرى مدهشة ، ولن نحاول هنا في هذا التمهيد أن نجاب عنها ، السؤال بعد الآخر ، ولكن سنحاول توضيح هدفنا من هذا الكتاب .

وضع هذا الكتاب بسبب ثلاث ملاحظات :

- إن إتقان لغة المؤول هو الطريقة الأفضل لفهم طريقة عمل الحاسب .
- بواسطة إتقان لغة المؤول ، مهما يكن ، سنستطيع التفكير بسهولة أكثر وإدراك ماذا يحدث عندما نعمل بلغة أكثر تطوراً ، والبحث عن الأخطاء سيكون أكثر سهولة .
- عند نزول الميكروبرومسور إلى الأسواق ، أليس من الأفضل إتقان هذه اللغة الموجودة على هذه الآلات الصغيرة ؟ مع الإشارة إلى أن المؤول يبقى الوسيلة الفضل لإنشاء وخلق المناهج الجديدة .

هكذا فلكتابنا هذا هدف تربوي . وهو ليس عبارة عن كتاب مساعد ومرجع في المعنى الذي نفهمه من المرجع المساعد الخاص بالمتج ، ولكنه عبارة عن مساعد كافٍ وكامل لفهم عمليات الإنشاء والبرمجة المهمة .

وهو موجه إلى أولئك الراغبين بفهم طريقة عمل الآلات التي يستعملونها . ولقد حاولنا الإجابة عن المسائل التي ستواجهنا ، وبشكل خاص لدى الطلاب الذين يرغبون بمعرفة لغة المؤول بعد معرفتهم بإحدى اللغات المتطورة . وهذا هو دور الفصل الأول من الكتاب الذي يحتوي على عرض لتركيب وطريقة عمل الحاسب ، وهذا العرض جرى من خلال تفكير بسيط يتعلق بآلة ذات استعمال كبير : الحاسب الجيبى . ولأجل هؤلاء

أيضاً قمنا بعرض مشاكل العنونة ، التقطيع ، تنقيح الأربطة (link editor) ، الشحن (loading) ، والإنقطاعات عند الإدخال والإخراج (I/O interruption) .

وهو موجه أيضاً الى كل من يرغب بالعمل بلغة المؤول ، إما على الآلة المعتمدة كمرجع وهي الحاسب IBM 370 ، أو على الحاسب الشخصي الميكروكمبيوتر . وهنا نؤكد بأن جميع لغات التأويل هي متشابهة بشكل نستطيع معه بعد معرفة مؤول معين أن نتكيف بسهولة للعمل على مؤول آخر بآلة أخرى ، ولهذا الهدف قمنا بإضافة مسائل بسيطة ، تجد التطبيق العملي لها على أغلب الحاسبات .

وفي النهاية ، هؤلاء الذين يعرفون المؤول ، قمنا بإثبات الإمكانات التي يقدمها التأويل المشروط وإستعمال الماكرو تعليمات (MACRO INSTRUCTIONS) . ونصائح هذا الكتاب التي تدور حول البرمجة الجيدة هي عبارة عن عناصر للتفكير يصبح في نهايتها البرنامج مختلفاً عن تلك المجموعة من التعليمات المبهمة كما في اللغة الثنائية . ومن الممكن إنشاء وتركيب برنامج مكتوب بلغة المؤول بشكل يصبح معه واضحاً كوضوح برنامج بلغة كويول .

لماذا جرى إختيار الحاسب IBM 370 ؟

- لأنها شاملة وعامة . وأكثر صيغ لغة المؤول العاملة عليها جرى إستعمالها وتطويرها من قبل جميع المنتجين والصانعين .

- لماضيها ومُستقبلها : إن المواصفات الخاصة بهذه اللغة والتي جهزت مع النظام IBM 360 ، قد جرت المحافظة عليها في الحاسبات IBM 370 وفي الأنظمة الجديدة من السلسلة 3000 و4000 إضافة إلى أغلب حاسبات IBM الجديدة .

### عموميات

#### 1. الآلة البسيطة

هذا الفصل الأول هو مخصص للمبتدئين . أما الذي يتمتع بمفاهيم كافية تتعلق بهيكل المكنة فيمكنه أن يبدأ دراسته من الفصل الثاني . إلا أننا نعتقد بأنه يعرض ويوضح النقاط الأساسية لعملية الفهم اللاحقة . وهو يعرف المصطلحات الأساسية المتعلقة بدورة تنفيذ تعليمات الآلة .

##### 1.1 . دراسة للآلة الحاسبة الصغيرة الجيبية

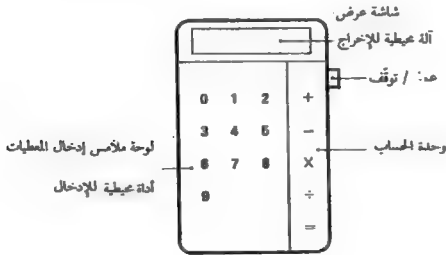
منذ النظرة الأولى ، تبدو الآلة الحاسبة الجيبية وكأنها مؤلفة من العناصر التالية :

- زر للعمل / ولوقت العمل .
- لوحة ملامس رقمية .
- شاشة للعرض .
- مجموعة من ملامس التحكم + ، - ، = ، ... .
- فلنقم بعملية حساب بسيطة ، القسمة مثلاً . عملية المعالجة متجري كما يلي :
- 1 - وضع الآلة الحاسبة في العمل .
- 2 - ادخال العدد الأول ( المقسوم ) وعرضه .
- 3 - ضغط الزر الخاص بالقسمة .
- 4 - إدخال العدد الثاني ( القاسم ) وعرضه .
- 5 - الضغط على الزر = ، وعرض النتيجة .
- 6 - إيقاف عمل الآلة الحاسبة .

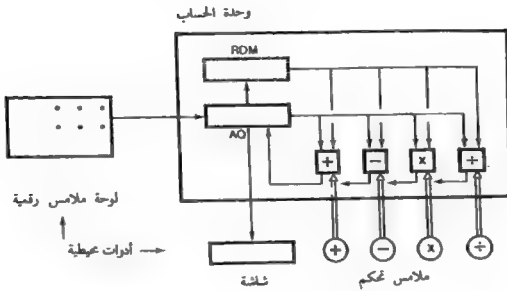
هذه السلسلة من العمليات تتطلب بعض الملاحظات :

- ترتيب العمليات هو مُحدّد وثابت ؛ لا يمكن عكس العمليات 2 و 4 .
- تتمتع مكتنا ، إضافة إلى الدالة حساب (Compute) ، بدالة ( مهمّة ) لإدخال المعطيات وبدالة لإخراج المعطيات ( العرض على الشاشة ) .
- عند إجراء العملية رقم 4 ، يختفي العدد المعروف على الشاشة ، قبل أن تتم عملية القسمة ( يجب أن نعطي الصلاحية للعملية بالضغط على الملمس = ) ، يجب إذن ،

وبشكل إلزامي ، أن تحتوي المكنة على ذاكرة يُخزّن فيها العدد الأول بانتظار نهاية إدخال القاسم . فلتعرض المخطط التوضيحي<sup>(1)</sup> :



1.1 مخطط



2.1 مخطط

(1) إن المخططات المعروضة في هذا الفصل لا تنمي تمثيل الدقة التكنولوجية ولكنها تعرض فقط الفئات الأساسية المفيدة للمبرمج .

هذا المخطط يُمَيِّز بين نوعين من الخطوط . الخطوط البسيطة (→) والتي تُناسب خطوط إنتقال المعطيات والخطوط المزدوجة(⇒) والتي تناسب خطوط تنقل الأوامر .

تعريفات :

نسَمي وحدة حساب مجموعة دارات الجمع والطرح ، ... تُخزَّن معطيات الحساب في المناطق RDM و AQ والتي تدعى مرافص (register) . المرصف RDM يُستخدم لتخزين العدد الأول الداخل إلى AQ للسماح بإدخال العدد الثاني .

نتيجة الحساب توضع دائماً في مرصف خاص AQ ولذلك نطلق عليه إسم مركم (Accumulator) . أما لوحة الملامس الرقمية وشاشة العرض فنطلق عليها الإسم : الأدوات المحيطة للإدخال والإخراج (I/O peripherals) .

## 2.1 . دراسة حاسبة جيبة مع ذاكرة

لننصف الى الحاسبة الجيبية مجموعة من خلايا الذاكرة التي سنطلق عليها الإسم : ذاكرة مركزية (Central memory) . كل خلية من الذاكرة ، وتدعى أيضاً كلمة - آية (machine word) ، يمكنها كالمراصف أن تحتوي على مخططات أو على نتائج الحساب . إلى كل خلية سربط عدداً محدداً يُدعى عنوان الخلية ويسمح بتمييز الخلايا فيما بينها . المؤثرات الأساسية ( + ، - ، ... ) هي عبارة عن مؤثرات ثنائية ( نقصد بذلك أنها تجري بين متأثرين (operators) ) . أحد المتأثرين يكون موجوداً في المرصف AQ والآخر في المرصف RDM ( مرصف معطيات الذاكرة ) . كما في الحاسبات البسيطة فإن النتيجة ستكون موجودة في AQ . يصبح من الضروري أن يكون بتصريفنا :

- نظام لإختيار العنوان الذي يؤمن الإتصال بين إحدى خلايا الذاكرة والمرصف RDM ؛

- دارتان إضافيتان للشحن والترتيب ، لشحن مضمون خلية من الذاكرة في المركم وترتيب مضمون المركم في عنوان معين . هكذا دارات هي موجودة على جميع الحاسبات الجيبية وتتمتع بخلية ذاكرة واحدة على الأقل . مخطط حاسبة كهذه هو ممثَّل على الشكل 3.1 .

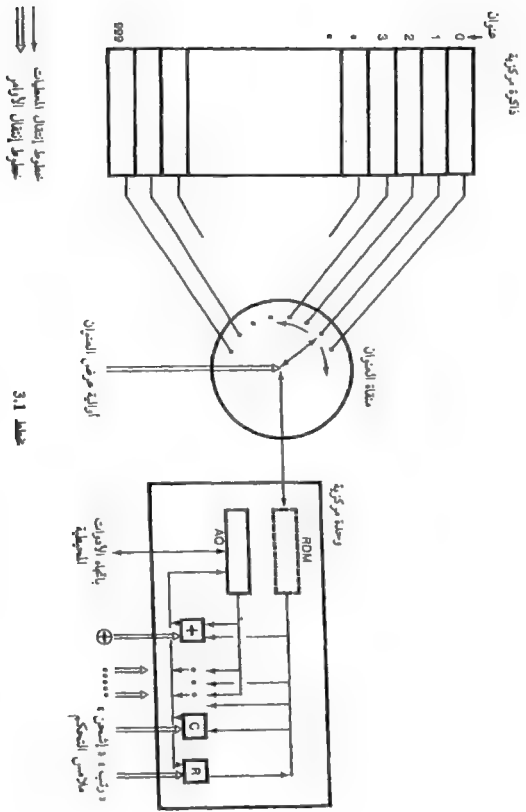
إن منقاة العنوان هي هنا موضحة بواسطة ملهاس دائري يؤمن الإتصال بين خلية من الذاكرة بعنوان معين ومضمون المرصف RDM . ويتعلق إنجاء إنتقال المعطيات بالمؤثر أو بالإشارة الحاسوبية المعتمدة .

مثال حول عملية حساب بسيطة .

لنفترض، إن الذاكرة تحتوي على المعطيات

التالية :

0	1	2	5
1	3	2	
2			



شكل 3.1

نرغب بجمع مضمون الخلية ذات العنوان 0 مع مضمون الخلية ذات العنوان 1 وبوضع النتيجة في العنوان 2 . فلنستعمل الترميز الكلاسيكي : (ALPHA) ، حيث ALPHA هي عبارة عن عنوان ، يشير الى مضمون الخلية ذات العنوان ALPHA . هكذا فإن (0) يعني هنا القيمة 125 . السهم سيؤدي إتمام انتقال المعطيات : AQ→(0) يعني تخزين مضمون الخلية ذات العنوان (0) في المرمز AQ ، أي تخزين العدد 125 في AQ .

لإجراء عملية الحساب يجب :

1 - تركيز متقاة العنوان على 0 والضغط على الزر «إشحن» ، مما يؤدي إلى تنفيذ العملية : AQ → (0) .

2 - تركيز متقاة العنوان على 1 والضغط على الزر + .

هذا يسمح بإجراء العملية  $AQ \rightarrow (1) + AQ$  . هكذا فإن هذه العملية يمكن تقسيمها إلى اثنتين .

أ)  $RDM \rightarrow (1)$

ب)  $AQ + RDM \rightarrow AQ$

3 - تركيز متقاة العنوان على 2 والضغط على الزر «تخزن» . هذا ما يسمح بتنفيذ العملية (2) :  $AQ \rightarrow$  .

في نهاية هذه العمليات ، تحتوي الخلية ذات العنوان 2 على العدد 157 . والمرصف AQ يحتوي على القيمة النهائية .

ملاحظات :

جميع عمليات الحساب تتم بين المرصاف AQ و RDM وليس من الذاكرة إلى الذاكرة . وهذا ما يؤدي إلى الحاجة إلى إجراء عملية شحن مسبقة للمرمز .

المرصاف هي إنفاً عبارة عن ذاكرة مرتبطة مباشرة بدارات الحساب .

للإشارة إلى مضمون خلايا الذاكرة سنستخدم الترميز (عنوان adresse) بشكل نستطيع معه تمييز العنوان عن مضمونه ، أي إسم «nom» الخلية وقيمتها . المرصاف المذكورة لا ترد داخل أهلة لأنه لا يوجد أي خلط ممكن بين المضمون والإسم : نعود دائماً إلى مضمون المرصف .

3.1 . من الحاسبة الصغيرة إلى الحاسب الكبير (الكومبيوتر)  
إن كل معالجة تتناول معطيات وتسلسلاً دقيقاً من الأفعال ، والأوامر على الملامس + ، - ، ... ونوع الحاسبة المعتمدة حتى الآن لا يسمح بتخزين معطيات المسألة .

الفرق الأكبر بين الحاسبة ذات الذاكرة والحاسب الكبير يكمن في كون الأخير :  
 - يُخزّن ليس فقط المعطيات ولكن الأوامر المطلوب إجراؤها على المعطيات .  
 - يتمتع بأولية لربط الأوامر التي ستسمح له بتنفيذ هذه الأوامر حسب الترتيب الواردة فيه . هكذا ، فذاكرة الحاسب المركزية (C.M) ستحتوي على معطيات المسألة وطريقة معالجتها للحصول على النتائج .

#### تعريفات :

في البداية ، سنعي كلمة أمر (Command) بالتعليلة (instruction) أو التعليلة الآلية (machine instruction) . ومجموعة التعليلات والمعطيات المرتبطة بها تؤلف البرنامج . أما الملامس + ، - . . . . فستختفي . ويصبح عندئذ من البديهي أن لا يعمل الحاسب إلا إذا كان البرنامج مسجلاً في ذاكرته المركزية .

#### 1.3.1 - هيكلية التعليلات الآلية

حسب المثل المذكور أعلاه في الفقرة 2.1 ، نستطيع أن نقول أن التعليلات الآلية هي مؤلفة من معلومتين :

- 1- رقم يدل على الدارة المعتمدة من اوحدة المركزية .
- 2- رقم يدل على عنوان المتأثر (Operand) .

إذا كانت التعليلة تعمل بمتأثرين ( الحالة + ، - ، . . . ) ، يكون المتأثر الأول مشحوناً مسبقاً في المرمك (ACC) . هاتان المعلومتان ستكونان موجودتين في كلمة من الذاكرة بشكل مكوّد رقمياً ، مثلاً حسب الطريقة التالية :



وستسمح أولية تكويد التعليلة ، التي ستقوم بتوضيحها لاحقاً ، بكشف ومعرفة الفعل المطلوب إجراؤه على المتأثر الموجود على العنوان المذكور في التعليلة .  
 مثلاً :

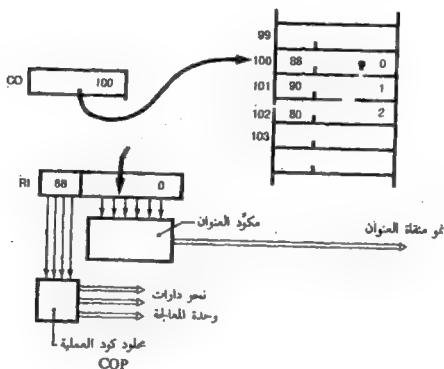
لنفترض بأن كود عملية الشحن COP هو 88 ، وإن كود الجمع هو 90 وكود التخزين هو 80 . فلنخزّن البرنامج الذي يقوم بجمع الخليتين 0 و 1 مع وضع النتيجة على العنوان 2 ، بدءاً من العنوان 100 . نحصل عندئذ على صورة الذاكرة التالية :

99			
100	8	8	0
101	9	0	1
102	8	0	2
103			

تنفيذ البرنامج يفترض ربطاً متتابعاً للتعليقات  
الموجودة ، بدءاً من العنوان 100 ثم 101 ، ...

### 2.3.1 . أولية معرفة وربط التعليقات

تحتوي الذاكرة على نوعين من المعلومات بطبيعة دلالية مختلفة . المعطيات والتعليقات . من الضروري معاينة ومعرفة الخلية التي تحتوي على التعليمة المطلوب تنفيذها . لهذا الهدف ، هناك مرصف خاص يسمى العدّاد الرئيسي الترتيبي (CO) أو عدّاد البرنامج program counter الذي سيحتوي في كل لحظة على العنوان التالي للتعليمة المطلوب تنفيذها . وبشكل خاص ، وفي البداية ، سيكون مشحوناً بعنوان أول تعليمة .



### 4.1 خط

منذ اللحظة التي يحتوي فيها CO على عنوان التعليمة ، فإن دورة التنفيذ تبدأ :

1 - إرسال التعليمة التي يشير إليها عداد البرنامج إلى مرصف التعليمة RI المرتبط بمكود العملية COP ومنقاة العنوان .

2 - تأكيد العنوان الذي يقوم بتركيز منقاة العنوان ، ومحمود ( يفك كود ) COP الذي يضع الدارة المناسبة من وحدة المعالجة في حالة العمل .

3 - تنفيذ العملية المطلوبة بواسطة وحدة المعالجة التي تصبح في طور العمل .  
خلال المرحلة الثانية لن يكون من الضروري أن يؤثر CO على التعليمة الموجودة في طور التنفيذ ، وخلال هذه المرحلة إذا تزداد قيمة عداد البرنامج CO واحداً (1) ليؤثر على التعليمة التالية المطلوب تنفيذها .

بعد تنفيذ التعليمة ، يعود الحاسب الى المرحلة الأولى بالقيمة الجديدة لعداد البرنامج CO وهذا يتتابع حتى نلتقي تعليمة خاصة بوقف البرنامج .

يقتضى أن نشير إلى مختلف مراحل التنفيذ هي مترابطة بواسطة نبضات ساعة داخلية .

المخطط 5.1 التالي يعرض لمختلف المهام التي درساها . وهو يشكل المخطط العملي للحاسب .

#### 4.1 - خلاصة حول المكنة البسيطة

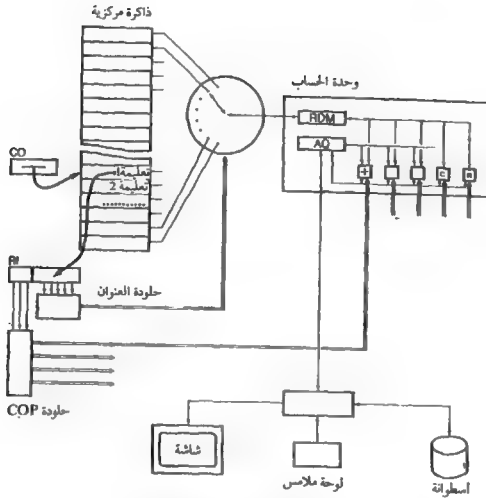
سنقوم بتوضيح الصيغ العملية للحاسب . إن جميع المكنات تستعمل هذه الآليات الأساسية ، إضافة إلى بعض التعديلات التي سندرسها عند الحاجة . فلنحاول الآن أن نستخلص بعض الملاحظات .

#### ملاحظة 1

المكنة المشروحة أعلاه هي مكنة « بعنوان بسيط » ، أي أن التعليمة الآلية لا تراجع سوى عنوان واحد وإذن متأثر واحد علني . في هذه الحالة ، لنفترض عدداً كبيراً من المؤثرات (operators) تستعمل متأثرين والنتيجة ، ذلك يعني أن أحد المتأثرين ثم النتيجة موجودان في المكنة . على بعض المكنات الأخرى قد نجد تعليمات تدعى « بعنوان مزدوج » .

---

(1) عندما تكون التعليمات ذات أطوال متغيرة ( حالة الحاسبات IBM 360/370 ) يتقدم العداد (CY) بمقدار طول التعليمة .



مخطط S.I. - الحاسب ، المخطط العملي

## ملاحظة 2

لا تحتوي مكتبتنا سوى مركب واحد . هناك حاسبات أكثر فعالية يمكن أن تحتوي على عدد من المرافيف التي تلعب دور المركب ( هذه هي حالة المكنة IBM 360/370 ) . سيكون من الضروري أن نشير ، من داخل التعليمية ، إلى رقم المرفص الذي نعمله كمركب .

## ملاحظة 3

لفترض ، كما في المخطط 3.1 ، أن ذاكرة المكنة تحتوي على 1000 خلية مرقمة من 0 إلى 999 . وهذا يعني أن :  
 1 - عدد البرنامج يحتوي على الأقل على ثلاثة مواقع عشرية تسمح له بمراجعة جميع عناوين الذاكرة المركزية ؛

2- ان حقل عنوان التعليم ، ونفس السبب ، يجب أن يسمح بتسجيل الأعداد من 0 إلى 999 .

#### ملاحظة 4

بعض التعليمات يمكن أن لا تُراجع بواسطة عنوان ما . تظهر هذه الحالة ، مثلاً ، عندما لا نستعمل سوى AQ (عكس إشارة AQ ، قصير AQ ، الإزاحة ، . . . ) . ولكن من الممكن ، عند الحاجة ، استعمال حقل العنوان لغايات أخرى . قد يحدث ، على بعض المكتبات ، أن يكون حقل العنوان مستعملاً ككود لعملية ثانوية ، مما يؤدي إلى زيادة عدد التعليمات بدون تعديل لحجم الحقل COP . أما الكود الثانوي فيُميز التعليمات الخاصة التي تنتمي إلى الفئة المحددة بواسطة الكود الرئيسي .

#### ملاحظة 5

الحجم ( هنا يقاس بعدد المواقع العشرية ) للحقل COP يُحدّد العدد الأقصى للدورات - أي للتعليمات الآلية - التي تراجع عنواناً وحيداً يمكن أن تحويه وحدة الحساب .

#### 5.1 . الحساب ، العرض الكلاسيكي

بعد هذا المدخل ، نعود إلى عرض أكثر كلاسيكية للحاسب . لقد جرت العادة أن تُميز بين الأعضاء التالية :

الوحدة المركزية وتحتوي :

- الوحدة الجبرية والمنطقية ( دارات عمليات ومرافق للحساب ) ،

وحدة التحكم وتتألف من :

- مرافق التحكم ،

- عداد البرنامج ،

- الساعة .

الذاكرة المركزية وتتألف من خلايا ( كلمات وبايتات ) معنونة ،

- أدوات محيطية تسمح بالإدخال والإخراج في الذاكرة المركزية للمعلومات ( برامج ومعطيات ) المخزنة على نواقل خارجية

فلنذكر البعض منها :

- قارئ البطاقات ، والمثبّات ، والطابعات ،

- بسّاطة الأشرطة ، الأسطوانات والطبول المغناطيسية ،

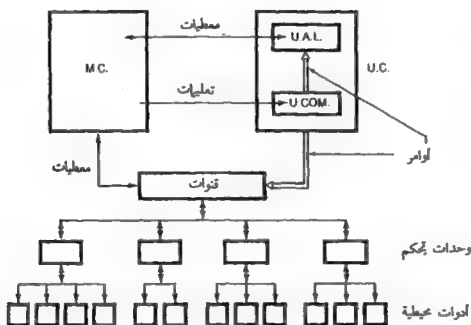
- لوحات ملاس ، - شاشات للعرض ،

- أدوات محيطية خاصة كراسم المنحنيات العاملة حسب النظام «off-line»

( الاشتغال المنزّل ) .

- القنوات أو وحدات التبادل . وهي عبارة عن الأعضاء التي ، تحت قيادة الوحدة المركزية ، تؤمن بشكل لا تزامني إنتقال المعطيات من الذاكرة المركزية إلى الأدوات المحيطية . هذه الأولوية تسمح بتحرير موارد الوحدة المركزية خلال الوقت ، نسبياً « الطويل » ، للإدخال والإخراج (I/O)<sup>(1)</sup> . التزامن بين الوحدة المركزية والقنوات (Channels) يتأمن بواسطة نظام الانقطاع الذي ستكلم عنه لاحقاً .

- وحدة المراقبة والتحكم (Control unit) وهي عبارة عن أجهزة وأدوات ، متكيفة مع كل نوع من المحيطات ، وتحقق عدداً من المهام الضرورية للإدخال والإخراج .



شكل 6.1

(1) أعضاء الإدخال - الإخراج هي أجهزة إلكتروميكانيكية تمثل إذن نوعاً من المقصور . إن قراءة بطاقة معينة قد تطول نحو 100 ميكروثانية في حين أن وقت تنفيذ تعليمية لا يدوم أكثر من الميكروثانية (م (10<sup>-6</sup> ثانية) )

## 2 تكويد المعلومات

الإستعمال الكثير للنظام العشري جعلنا معتادين عليه ، وهذا الإعتياد جعل البعض يخشى من إستعمال نظام آخر للترقيم . ولكن تكنولوجيا الحاسبات تفرض علينا دراسة أنظمة تكويد مختلفة . يجب أن نشير إلى أن التمثيل الثنائي للمعلومات في المكنة لا يحمل أي تعديل لصيغة العمل المشروحة في الفصل الأول ، وهذا من الأسباب التي جعلتنا لا نبداً الكتاب بهذا الفصل ، راجين أن يكون عرضنا أكثر وضوحاً .

يتألف نظام التكويد من مجموعة قواعد التحويل التي تسمح بالعبور من تمثيل للمعلومات ( نص فرنسي مثلاً ) إلى ترميز آخر ( نص بكود مورس . . ) والعكس بالعكس .

الترميز الثنائي هو مفروض لأنه يسمح بتمثيل بسيط لمضمون الذاكرة والمراسف في الحاسب<sup>(1)</sup> . ويبدو أنه لترميز عدد  $n$  من حالات صمام كهربائي ، مولّع أو مطفأ ، فإن التمثيل الثنائي هو الأبسط باعتياد الاتفاق التالي :

1 - حالة « الضوء »

0 - حالة الإنطفاء



إذاً يرمز إلى الحالة بواسطة :

(1) دون الدخول في التفاصيل التكنولوجية ، تمثل المعلومات داخل الآلة بواسطة عناصر تمتلك حالتين فيزيائيتين مختلفتين .

قد نلاحظ أن مجموعة من صمامين يمكن أن تكون موجودة في عدد  $4=2^2$  من الحالات المختلفة التي نرمز إليها على الشكل التالي :

0	0	حالة «0»
0	1	حالة «1»
1	0	حالة «2»
1	1	حالة «3»

ولكن بإمكاننا تكويد :

الحالة «0» : الصمامان هما في حالة الإنطفاء

الحالة «1» : الصمام اليسار هو مطفأ ، والصمام اليمين مؤلف ، الخ

وبشكل عام ، فإن مجموعة من  $n$  من الصمامات يمكن أن تكون موجودة في  $2^n$  حالة مختلفة . يجب تقريب ذلك من الفعل الذي يسمح بواسطة  $n$  رقم ثنائي بأن نمد من 0 إلى  $2^n - 1$  .

1.2 . أنظمة الترقيم :

لو افترضنا أن  $\mathbb{B}$  تمثل مجموعة الرموز المستعملة لتحديد عدد بالقاعدة  $\mathbb{B}$  ، فإن العدد الحقيقي  $R$  يكتب على الشكل التالي :

$$\frac{a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots}{\text{القسم الصحيح} \quad \text{القسم العشري}}$$

وقيمة هي :

$$R = \frac{a_n B^n + a_{n-1} B^{n-1} + \dots + a_0 B^0}{\text{القسم الصحيح}} + \frac{a_{-1} B^{-1} + a_{-2} B^{-2} + \dots}{\text{القسم العشري}}$$

وفي النظام العشري فإن المجموعة  $\mathbb{B}$  تتألف من الرموز :

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

وفي الثنائي : 0 و 1 .

وفي النظام الثنائي : 0, 1, 2, 3, 4, 5, 6, 7 .

وفي النظام السادس عشري (16) : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,

A, B, C, D, E, F

إنَّ أسَّات القاعدة  $B^0$  ،  $B^1$  ،  $B^2$  ، ... ،  $B^{-1}$  ، ... تدعى أوزان الأرقام .  
الجدول 1.2 يعطي قيم بعض الأوزان بالنظام العشري :

القاعدة	$B^3$	$B^2$	$B^1$	$B^0$	$B^{-1}$	$B^{-2}$
10	1000	100	10	1	0,1	0,01
2	8	4	2	1	0,5	0,25
8	512	64	8	1	0,125	0,015625
16	4096	256	16	1	0,0625	0,00390625

## جدول 1.2

هكذا فالعدد 13 في القاعدة 10 يعادل  $(3.10^0 + 1.10^1)$   
ويكتب على الشكل التالي :  $(1.2^0 + 0.2^1 + 1.2^2 + 1.2^3)$  في النظام الثنائي .

15 في النظام الثنائي :  $1.8^1 + 5.8^0$   
D في النظام السادس عشري :  $(13.16^0 \text{ أي } D.16^0)$   
والعدد 0,75 في النظام العشري :  $(5.10^{-2} + 7.10^{-1})$   
يكتب : 0,11 في النظام الثنائي :  $(1.2^{-2} + 1.2^{-1})$   
0,6 في النظام الثنائي :  $(6.8^{-1})$  .  
و 0,C في النظام السادس عشري :  $C.16^{-1}$  أي  $12.16^{-1}$  .

وفي المكنة ، تمثَّل الأعداد بشكل مكوَّد ثنائيًا . ويمكن أن يحتاج عدد عشري كسري إلى سلسلة طويلة ، أو لا نهائية ، من 0 و 1 . وبما أن الذاكرة والمرافق لها أبعاد محدَّدة عند تصميم المكنة ، لذا ، فقد يحدث تحويل عشري / ثنائي عند الحساب ، أو قد يحدث بتر لقسم من المعلومات مما يؤدي إلى فقدان الدقة في الحساب . وهذه من المشاكل التي يجب الانتباه إليها ولذا من الواجب القيام بعدد كبير من الحسابات التكرارية .

من المهم أن نلاحظ ، أنه عند إزاحة الفاصلة ٣ موقع لجهة اليسار أو لجهة اليمين فإن هذا يؤدي إلى ضرب العدد أو قسمته على ١٠٠٠ مثلاً : 13,75 يمثل بواسطة العدد 1101,11 في النظام الثنائي ، ولكن 11011,1 يعادل 27,5 و 110,111 يعادل 6,٨75 .

ثنائي	سادس عشري	ثنائي	عشري
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

## جدول 2.2

### 2.2 . تغيير القاعدة

سنترك للقارئ أن يعود للمراجع إذا رغب بذلك . وسنذكر ، بواسطة بعض الأمثلة ، إن التحويلات الثنائية / الثنائية والثنائية / السادس عشرية هي مترابطة لأن القواعد 8 و16 هي عبارة عن أسس صحيحة للقاعدة 2 .

ينقلب العدد الثنائي إلى سادس عشري بدءاً من كل جهة من موقع الفاصلة . ويتقطع العدد إلى أقسام مؤلفة من أربعة أرقام ثنائية أو بتات<sup>(1)</sup> ويتأويل كل قسم :

$$\begin{array}{r} 1001101011,11001 \\ \hline 2 \quad 6 \quad B \quad C \quad 8 \end{array} \quad \begin{array}{l} \text{ثنائي} \\ \text{سادس عشري} \end{array}$$

الرقم الأخير «8» نحصل عليه بتوسيع الرقم 1 بوضع أصفار لجهة اليمين . التحويل الثنائي / الثنائي يتم بتقطيع العدد الثنائي إلى أقسام مؤلفة من ثلاثة أرقام . نحصل عندها على 1153, 62 في النظام الثنائي . التحويل المعاكس هو بلعبي .

### 3.2 . الفائلة من النظامين السادس عشري والثنائي

سنرى أن كل كلمة آلية هي مكونة من عدد متحول ، يتعلّق بالحاسب ، من العناصر التي تدعى بتات<sup>(1)</sup> (bit) . كل عنصر يمكن أن يكون موجوداً ، كما هي الحالة

(1) من BFT وهو اختصار للمصطلح الأمريكي Binary digit ، أي رقم ثنائي .

بالنسبة للصيَّام ، في واحدة من حالتين فيزيائيتين ، لذا يصبح من الطبيعي ترميز حالة البتة بواسطة 0 أو 1 ومضمون الكلمة - الآلية ؛ ليس كما في الفصل الأول بواسطة رقم عشري ، بل بواسطة سلسلة من الأرقام 0 أو 1 ، ويمكن تفسير مجموعة البتات كعدد تمثِّل في النظام الثنائي .

الأحجام ، المحددة بعدد البتات ، للكليات - الآلية التي نلتقيها عادة في الحاسبات هي بطول 8 ( الميكروبروسسور ) ، 16 ، 24 ، 32 ( IBM 360/370 ) ، 36 ، 48 و 60 بتة . عند تمثيل مضمون كلمة - ذاكرة على ورقة فهذا يتطلب من 16 إلى 60 رمزاً . التمثيل السادس عشري والثاني يظهران إذن مفيدتين مهمَّين كثيراً لأنها يُعَسَّبان على 4 أو على 3 عدد الرموز المطلوب كتابتها وذلك مع المحافظة على إمكانية تحويلها فوراً إلى النظام الثنائي . ولكن النسخ اليدوي لعدد محدد بالنظام السادس عشري هو منبع لعدد أقل من الأخطاء منه في حال كتابته في النظام الثنائي . لذلك فللقارىء فائدة من الإعتياد على هذا النوع من التمثيل المعتمد لتمثيل المعلومات في الذاكرة .

4.2 . الحساب في النظامين الثنائي والسادس عشري  
لن نقوم سوى بإعطاء بعض الأمثلة التي يجب أن تسمح للقارىء بإجراء بعض العمليات البسيطة بالجمع والطرح .

في النظام الثنائي :

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ +0 \quad +1 \quad +0 \quad +1 \\ \hline 0 \quad 1 \quad 1 \quad 10 \end{array}$$

$$\begin{array}{r} 1101 \\ +111 \\ \hline 10100 \end{array} \quad \text{مثلاً :}$$

$$\begin{array}{r} D \\ +7 \\ \hline 14 \end{array} \quad \text{في النظام السادس عشري :}$$

في النظام السادس عشري من العملي تحويل كل رقم إلى النظام العشري ، وإجراء العملية في هذا النظام ومن ثم تحويل النتيجة . مثلاً :  
نضع  $D_{16}=13_{10}$  ,  $7_{16}=7_{10}$  ,  $13+7=20_{10}=16+4$  : إلخ :

$$\begin{array}{r} 3F2 \\ +1A4 \\ \hline 596 \end{array} \quad \begin{array}{r} 3F2 \\ -1A4 \\ \hline 24E \end{array}$$

بنفس الطريقة نقوم بإجراء الطرح  $4 - 2 = 2$  تصبح  $4 - 16 + 2$  أي E وبالبداية 1 . . .  
حسب نفس الصيغة سنستطيع إجراء الحساب في النظام الثنائي . وباستطاعة  
القارئ أن يتمعّن بوجود الأمثلة المطلة في نهاية الفصل .

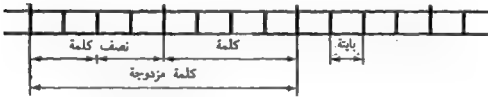
## 5.2 . التمثيل الداخلي للمعطيات

### 1.5.2 . الذاكرة

حتى هذا الوقت إعتبرنا إن الذاكرة هي مؤلفة من خلايا مرقّعة بدءاً من 0 ،  
الخلية هي الكلمة - الآلية والعناوين هي عناوين الكلمات .

سنقوم بتحديد الأشياء . المكنتات IBM 360/370 تتمتع بكلمة - آلية من 32 بتة  
مرقّعة من اليسار إلى اليمين من 0 إلى 31 . تُقسّم الكلمة إلى أربع بايتات ( تشكيلة من  
8 بتات ) . والبايتة هي قابلة للنعونة . ستكلّم عن الذاكرة المنعونة بالسّيات (وسمى  
إن السمة قابلة للتمثيل بواسطة 8 بتات ) مقابلة مع بعض المكنتات حيث الذاكرة معنونة  
بالكلمات . عنوان الكلمة هو إذاً عنوان البايتة الأولى من الكلمة . في النهاية نوجز ما  
يلي :

- جهات النصف كلمت هي بعناوين مزدوجة ؛
  - جهات الكلمات هي بعناوين قابلة للقسمة على أربعة ؛
  - جهات الكلمات المزدوجة تتمتع بعناوين قابلة للقسمة على 8 ؛
- ومع إن الذاكرة هي قابلة للنعونة في مستوى البايتة ، يجب السهر على المحافظة  
على هذا التقسيم للمعطيات المثلثة بواسطة نصف كلمة ، كلمة ، أو كلمة مزدوجة .



شكل 3.2

### 2.5.2 . تمثيل المعطيات اللارقمية

إمكاننا تكويد نوعين من المعلومات في الذاكرة : المعطيات الرقمية والتي هي  
عبارة عن تشكيلات ثنائية مرتبطة بمعنى رقمي ، والمعطيات من نوع سيات والمعالجة  
كوححدات غير رقمية .

لقد كان من الملائم عند تصوّر مكنات IBM 360/370 ، توكيد السّيات بواسطة 8 بتات . هذا النظام يسمح بتوكيد  $2^8$  ، أي ما مجموعه 256 كوداً مختلفاً . هذا التصوّر هو واسع الإنتشار ، ولكن هناك مكنات أخرى تستعمل توكيد السّيات بواسطة 6 بتات مُحمّد مجموعة السّيات المتوفّرة بالعدد 64 سمة .

قد يبدو لنا مفاجئاً اعتياد كود لتمثيل السّيات بواسطة 8 بتات . فلنلاحظ ببساطة إن هذا النظام يسمح لنا بالحصول على ألفباء واسعة تحتوي على السّيات الكبيرة ، والصغيرة ، والسّيات العشرية وبعض السّيات الخاصة ، كإشارات العمليات ، وعلامات الوقف ، والفسحة ، الخ .

الكود الداخلي لتمثيل السّيات ، والمستعمل على المكنات IBM 360/370 هو EBCDIC (Extended Binary Coded Decimal Interchange Code) . يُرمز إلى الحرف A بواسطة الكود 11000001 ، أي C1 بالترميز السادس عشري . ويكوّد الحرف «B» بواسطة C2 وهكذا دواليك . لائحة الأكواد موجودة في الملحق .

مثلاً : لنفترض إن مضمون حيّز الذاكرة هو التالي :

0	0	C	1	E	2	E	2	C	5	D	4	C	2	D	3	C	5	E	4	D	9	4	0	F	3	F	7	F	0	D	0
100								104								108								112							

تأويل هذه السلسلة من 14 بايتة ، والتي تبدأ بالعنوان 100 ، هو حسب الكود «ASSEMBLER 370» .

نشر إلى وجود علاقة تراتبية بين القيم الثنائية المستعملة للتوكيد :

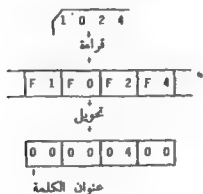
$$40 < C1 < C2 < \dots < F0 < F1 < \dots < F9$$

وهذا يمكن أن يترجم بواسطة :

كود الأرقام > ... > كود B > كود A > كود القسمة .

هذه الخصوصية هي مستعملة للترتيب الأبجدي .

يجب أن نُميّز بين التمثيل الأبجدي والتمثيل الرقمي . المثال التالي يوضح لنا التحويل المعتمد لمعطى مقروء من البطاقة ومحوّل إلى ثنائي .



نقل خارجي

ذاكرة (حيّز إدخال)

تمثيل بطريقة السّيات

ذاكرة (متحوّلة مؤشرة في)

FORTRAN في لائحة الأمر (READ) .

تمثيل بطريقة الفاصلة الثابتة

التمثيل السبائي يُقال عنه أيضاً « القابل للتفجح » لأنه ضمن هذا الشكل يجب أن تكون المعلومات موجودة قبل أن تستلمها الطابعة لطبعها .

### 3.5.2 . تمثيل المعطيات الرقمية

المعتادون على لغة فورتران يعلمون أن المتحولة أو الثابتة يجب أن تُمثل دائماً في المكتبة بواسطة كلمة ( أو كلمة مزدوجة عندما يكون الحيزُ مَصْرُوحاً عنه بدقة مزدوجة ) . ويعلمون أيضاً إن هذه اللغة تستعمل نوعين أساسيين من التمثيل الداخلي للمعطيات الرقمية : النوع الصحيح (integer) والنوع العائم (real) .

لما المعتادون على لغة كويول فلا يجهلون أن الحسابات الجارية بهذه اللغة تتم بواسطة تمثيل مجهول من لغة فورتران : التمثيل العشري المتراس . سنجد هذه الطرق الأربع في توكيد الأعداد في مستوى المكتبة : الطريقة «الفاصلة الثابتة» (fixed point) « صحيح بلغة فورتران » ، والعالم البسيط والعائم الموسع والصيغة العشرية المتراسة . نشر إلى أن مع كل نوع من هذه التمثيلات تتلام مجموعة من المؤثرات ( دارات الكترونية ، + ، - ، ... ) ، صالحة للعمل بهذه التشكيلات الثنائية . وفي النتيجة فإن المكتبات تحتوي على أربع مجموعات من التعليمات الجبرية .

#### أ - التمثيل بفاصلة ثابتة

بهذه التسمية يجب أن نفهم « فاصلة ثابتة إتفاقياً » . هكذا ، فالفاصلة ، عنصر أساسي من قيمة العدد ، لا تظهر أبداً في التمثيل الداخلي للعدد في الذاكرة . ولقد لاحظنا ( في الفقرة 1.2 ) أن التشكيلات الثنائية المعتمدة لـ  $n$  ،  $2n$  و  $n/2$  لا تختلف إلا بواسطة موقع الفاصلة ، لذا ، فإن 1001 يمكن أن تُمثل القيمة 9 إذا اعتبرنا أن الفاصلة موجودة لجهة اليمين ، أو 0.5625 إذا اعتبرنا أن الفاصلة موجودة في أقصى اليسار النظام IBM 360/370 يفترض الفاصلة موضوعة لجهة اليمين . وللتأكد من ذلك يكفي ملاحظة التعليقات التي تسمح بجمع المعطيات بطول مختلف ( كلمة أو نصف كلمة ) . إن عملية التسطير للمعلومات تتم لجهة اليمين . هذا التمثيل هو إذاً التمثيل الصحيح . وهناك بعض المصممين الآخرين الذين اعتمدوا الإتفاق المعاكس ، أي الفاصلة لجهة اليسار .

تؤكدُ الأعداد حسب النظام الثنائي في كلمة - آلة . البتة ذات الوزن الأكبر ( البتة الموجودة لجهة اليسار ) ترمز إلى الإشارة الحسائية . إذا كانت تساوي 0 ، يكون العدد إيجابياً ، أما إذا كانت تعادل 1 فمعنى ذلك أن العدد هو سلبى .

بواسطة  $n$  بتة باستطاعتنا تعداد من 0 حتى  $2^{n-1}$  . وإذا حجزنا بتة للإشارة فسيكون بإمكاننا تمثيل الأعداد الصحيحة I بحيث إن :

$$-2^{n-1} \leq I \leq 2^{n-1}-1$$

إذا كانت  $n = 16$  :  $n = 32767 + I \leq 32768 -$

تمثيل الأعداد الإيجابية لا يفترض أية مشكلة ، والتأويل العشري نحصل عليه بضرب كل بته بالوزن المعتمد للموقع . وفي المقابل يجب أن نعتد إتفاقاً جديداً للأعداد السلبية .

تمثيل الإشارة والقيمة المطلقة

الفكرة التي نخطط لنا تقوم على إعتبار البتة ذات الوزن الأقوى ترمز إلى الإشارة والباقي يرمز إلى القيمة المطلقة للعدد . حسب هذا الإتفاق ، الممثل بأربع بتات :

$$\begin{array}{rcl} 0101 & : & + 5 \\ 1101 & : & - 5 \\ \hline 10010 & : & \text{نتيجة الجمع} \end{array}$$

هذه النتيجة هي ليست حقيقية .

هذا التمثيل يحتم علينا إذاً ، للحصول على النتيجة الصحيحة ، أن نفحص الإشارات المرتبطة بالمتغيرات قبل إجراء العمليات . لا يجب معالجة الأعداد السلبية والإيجابية بنفس الطريقة . يمكن للقارئ أن يقتنع بأن إعتداد هذه الصيغة يحتم علينا إعتداد منطق الكترول أكثر تعقيداً . وقد جرى التحلي عنه اليوم .

التمثيل المدعو مُكمّل 1 (1 Complement)

عكس العدد (أو ضده) . نحصل عليه بأخذ عكس كل بته . بما فيها بته الإشارة . هكذا :

$$\begin{array}{rcl} 0101 & : & + 5 \\ 1010 & : & - 5 \\ \hline 1111 & : & \end{array}$$

وبنتيجة الجمع نحصل على

أي ، أن مُكمّل 1 هو 0000

من الممكن إعتبار إن هذا النوع من التمثيل يؤدي إلى إدخال 0 إيجابي وصفر سلمي . المسائل المطروحة في نهاية الفصل تشرح سينات هذه الاتفاقات وفوائد الاتفاقات اللاحقة

التمثيل المدعو مُكمّل إلى 2 (Two complement)

هو التمثيل المعتمد على الكائنات IBM 360/370 . يُمثل كل عدد سلمي بواسطة المُكمّل إلى  $2^n$  لعكس العدد . ولو افترضنا إن  $X$  هو العدد ، وأن  $\bar{X}$  هو مُكمّل العدد  $X$  إلى  $2^n$  ، نحصل إذاً على العلاقة التالية  $X + \bar{X} = 2^n$  . الإتفاق حول الإشارة هو كالسابق . ونشير إلى أن المعطيات الرقمية هي مكوّنة بأطوال ثابتة ، هي الكليات -

الآلية . وللمكنات IBM 360/370 ، n تعادل 32 . ولتسهيل العمل ، فإننا سنعالج مسائل نعمل بأربع أو ثمان بتات .

وبالتكويد بواسطة أربع بتات ، حيث البتة اليسرى هي بتة الإشارة ، فإن كود العدد 5- هو المعادل الثنائي لـ  $2^4 - 5 = 11$  إذن :

$$\begin{array}{r} +5 \\ -5 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0101 \\ 1011 \\ \hline 10000 \end{array}$$

ويإهمال الحاصل بعد موقع الإشارة نحصل على صفر .  
الطريقة للحصول على المكمل إلى 2 لعدد ما تكمن بتكملة العدد إلى 1 وبعد ذلك إضافة 1 إليه . تتم العمليات على جميع البتات بما فيها بتة الإشارة .  
مثلاً :

$$\begin{array}{r} 0101 \\ 1010 \\ +1 \\ \hline 1011 \end{array} \qquad \begin{array}{l} +5 \\ 1 \text{ مكمل إلى } \\ +1 \end{array}$$

فلنلاحظ إنه إذا كنا نعمل على عدد ثنائي مُمثّل بالتزقيم السادس عشري ، فإن المكمل إلى  $2^n$  يصبح مُكملاً إلى  $16^n$  سنحصل على التمثيل السادس عشري للعدد المعكوس بتكملة كل رقم إلى F وإضافة 1 .  
مثلاً : على ثمان بتات :

$$\begin{array}{r} 0100 \ 1101 \rightarrow 4D \\ + \\ B2 \\ +1 \\ \hline 1011 \ 0011 \leftarrow B3 \end{array}$$

انتقال العدد ، المُمثّل بواسطة 16 بتة ، في مرصيف بطول 32 بتة سيتم بواسطة انتقال بسيط إلى اليسار للبتة ذات الوزن الأكبر :

$$\begin{array}{l} 0A1C \rightarrow 0000 \ 0A1C \\ B0D3 \rightarrow FFFF \ B0D3 \end{array}$$

حالة الفيض عن السعة (Over flow) ، يمكن أن تحدث عند إجراء عملية معينة وذلك عندما يكون كلا المتأثرين بنفس الإشارة والنتيجة تصبح بإشارة معاكسة .  
 نلاحظ بعض الأمثلة على معطيات ممثلة بواسطة أربع بتات . مجموعة الأعداد القابلة للتمثيل هي :

1111	-1	0000	0
1110	-2	0001	+1
1101	-3	0010	+2
1100	-4	0011	+3
1011	-5	0100	+4
1010	-6	0101	+5
1001	-7	0110	+6
1000	-8	0111	+7

بالإمكان إكمال المرحّل اليسار بدءاً من موقع الإشارة ، إذا كان كلا المتأثرين بنفس الإشارة ، والنتيجة بنفس الإشارة .  
 وجود المرحّل ، ويُسمى (Carry) في المصطلحات الأنكلوسكسونية ، ليس هو إشارة خطأ في الحساب .  
 سنلاحظ في النهاية إن العدد الأصغر القابل للتمثيل هو  $2^{-1}$  -2 والأكبر هو  $2^{+1}$  -1 وإن الطرح يمكن أن يتم بواسطة الجمع إلى مكمّل 2 .

نتيجة		مرحّل مفقود	عشري
خطأ ODC (1)	$\begin{array}{r} 0\ 111 \\ 0\ 111 \\ \hline 1\ 110 \end{array}$		$\begin{array}{r} +7 \\ +7 \\ \hline 14 \end{array}$
خطأ ODC (1)	$\begin{array}{r} 0\ 100 \\ 0\ 101 \\ \hline 1\ 001 \end{array}$		$\begin{array}{r} +4 \\ +5 \\ \hline 9 \end{array}$
صحيح	$\begin{array}{r} 0\ 100 \\ 1\ 011 \\ \hline 1\ 111 \end{array}$		$\begin{array}{r} +4 \\ -5 \\ \hline -1 \end{array}$
خطأ ODC (1)	$\begin{array}{r} 1\ 100 \\ 1\ 011 \\ \hline 0\ 111 \end{array}$	1	$\begin{array}{r} -4 \\ -5 \\ \hline -9 \end{array}$
صحيح	$\begin{array}{r} 1\ 101 \\ 0\ 011 \\ \hline 0\ 000 \end{array}$	1	$\begin{array}{r} -3 \\ +3 \\ \hline 0 \end{array}$

ODC (1) : الفيض عن السعة (over flow)

## ب - التمثيل بفاصلة متحركة

الحساب العلمي يستعمل عادة أعداداً بأحجام كبيرة جداً أو صغيرة جداً ولكن ممثلة بواسطة عدد محدد من الأرقام . النوع فاصلة ثابتة لا يسمح بالتمثيل البسيط لهذه الأعداد ، ولذلك إعتدنا طريقة أخرى في التكويد المركب من قسمين :

- المميّزة ( الأسّ المعيّن ) التي تعطي الحجم .
- القسم العشري ( mantisse ) الذي يحدد الأرقام ذات الأوزان الكبرى .

هكذا فإمكاننا تمثيل العدد على الشكل التالي :

S.M.B<sup>c</sup>

حيث S هي الإشارة ، M القسم العشري ( mantisse ) ، B عدد ثابت ( 2 ، 10 ، أو 16 حسب المكنة ) ، C هي الأسّ المعيّن .

كما في الفاصلة الثابتة ، فإن الفاصلة لا تظهر في التكويد الداخلي ولكنها توضع عادة إلى يمين أو إلى يسار القسم العشري M . هكذا ، فلنظام بقاعدة B=10 ، يكتب العدد 37.52 على الشكل التالي<sup>(1)</sup> :

1 - الفاصلة لجهة يمين القسم العشري .

(1)  $37520 \cdot 10^{-3}$



(2)  $3752 \cdot 10^{-2}$



2 - الفاصلة إلى يسار القسم العشري ،

(3)  $0.00003752 \cdot 10^7$



(4)  $0.3752 \cdot 10^2$



نلاحظ ، في الحالة التي تكون فيها الفاصلة موجودة إلى يسار القسم العشري ، إن التمثيل (4) يعطي عدداً أكبر من الأرقام ذات المعنى (Significants digits) من التمثيل

(1) Δ : رمز يشير إلى موقع الفاصلة .

(3) ، في الحالة التي يكون فيها عدد الأرقام المحجوزة للقسم العشري ثابتاً . التمثيل (4) يُدعى موحد التنظيم المعايير (normalized) . وهو يتناسب مع حصر الأرقام ذات المعنى من القسم العشري لجهة اليسار . هذا التمثيل يسمح بأكثر دقة ممكنة .  
من الممكن أن نغير من تمثيل معين إلى تمثيل معايير آخر بواسطة إزاحة الأرقام وتعديل الأس .

إذا كانت  $B = 10$  ، فإن الإزاحة إلى اليسار لموقع رقم يؤدي إلى تنقيص الأس المعين 1 .

أما إذا كانت  $B = 16$  ، فإن الإزاحة إلى اليسار لرقم سادس عشري من القسم العشري سيؤدي إلى تنقيص 1 من الأس المعين . وهكذا سيكون العدد ممثلاً بشكل معايير عندما لا يكون الرقم السادس عشري ذو الوزن الأكبر من القسم العشري صفراً . سنشير إلى أن الإزاحة لموقع سادس عشري يُترجم إزاحة أربعة مواقع ثنائية . على الحاسبات IBM 360/370 :

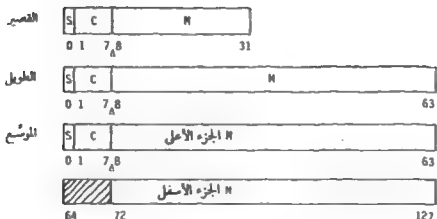
- الإشارة S من العدد هي مكوّنة على بته واحدة (  $0 = +$  ،  $1 = -$  ) ؛
- القاعدة B هي 16 ؛
- يفترض أن تكون الفاصلة إلى يسار القسم العشري الذي يُمثّل عدداً أصغر من 1 .
- العدد الثنائي المكوّن في الحيز C بطول 7 بتات والمحفوظ للأس المعين ، لا يُمثّل أبداً قيمة الأس المعين E إلا 16 ولكن :

$$C = 64_{10} + E$$

لذا فهناك مشكلة في توكيد إشارة الأس كي نحصل على قوى سلبية وإيجابية للقاعدة B بدلاً من اعتماد ترميز شبيه بالمكمل إلى 2 ، لقد جرى إختيار اعتماد القوة صفر في التوكيد المناسب للقيمة الوسطية للأعداد القصوى 0 و  $2^7 - 1$  أي  $64_{10}$  أو  $40_{16}$  أو  $1000000_2$  . هكذا ، عندما تكون  $C = 64_{10}$  فإن قيمة العدد هي  $S.16^0.M$  ، وعندما تكون  $C > 64$  (  $C = 64 + E$  ) فإن قيمة العدد هي  $C.S.16^E.M$  متغيرة من 0 إلى 127 وبالتالي  $-64 \leq E \leq +63$  .

للحصول على E يكفي ، في النظام السادس عشري ، أن نطرح  $40_{16}$  :  $40_{16}$  تناسب  $E = 6$  و  $3F_{16}$  تناسب  $E = -1$  .

يوجد على الحاسبات IBM 370 ثلاثة أشكال بفاصلة متحركة . الأعداد بالفاصلة المتحركة الصغيرة تحتل كلمة - آلية ، والأعداد الطويلة تحتل كلمتين - آليتين ، والأعداد الموسّعة تشغل أربع كلمات . الشكل الأخير هو غير موجود على المكتات 360 .



#### خط 4.2

الأشكال الثلاثة تسمح بتكويد أعداد بنفس الحجم . وتختلف بواسطة عدد الأرقام ذات المعنى التي تقلعها . العدد P هو :  
بالشكل القصير :

$$16^{-63} < P < (1 - 16^{-63}).16^{63}$$

7 أرقام عشرية ذات معنى

$$16^{-63} < P < (1 - 16^{-14}).16^{63}$$

- بالشكل الطويل

16 رقماً عشرياً ذا معنى .

$$16^{-63} < P < (1 - 16^{-28}).16^{63}$$

- بالشكل الموسع

33 رقماً عشرياً ذا معنى .

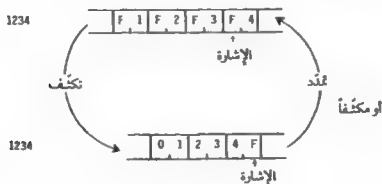
وفي الحالات الثلاث يكون معنا تقريباً :  $5,4.10^{-79} < P < 7,2.10^{75}$

أما الحسابات بواسطة هذه الطرق في التمثيل فقد تؤدي إلى فيض عن السعة (Overflow) عندما نحصل على قيم كبيرة جداً أو صغيرة وتدعى Overflow أو Underflow للأعداد بالفاصلة المتحركة .

مثلاً : التمثيل بفاصلة متحركة

C 2 1 9 0 0 0 0	-25	$-(1.16^{-1} + 9.16^{-2}) 16^2$
C 1 1 0 0 0 0 0	-1	$-(1.16^{-1}) 16^1$
0 0 0 0 0 0 0	0	$0 \cdot 16^{-64}$

ج - التمثيل العشري  
يمكن أن يتم تمثيل العدد بواسطة النظام العشري المكدّد ثنائياً (DCB) موسّعاً ،  
أي على شكل سمات .



هذا التمثيل ، الواسع الإنتشار في الإدارة ، هو أقل « ترصاً » من سوابقه . لا يوجد أي طول ضمني لها : توضع المعطيات بداخل بايتات . منرى ان التعليقات PACK و UNPK تسمح بالعبور من شكل إلى آخر .

الفاصلة ، كما رأينا ، ليست مُمثلة . وإن تنظيم موقعها والاصطفاف المحتمل المناسب يقع على عاتق المبرمج . ونشير إلى المواقع المختلفة للإشارة . القيم السادس عشرية A, C, F, E يجري تأويلها وكأنها « + » . أمّا D فيؤوّلان وكأنها « - » .

## تمارين

تمرين 1.2 - غيّر إلى النظام الثنائي والسادس عشري ، الأعداد العشرية التالية :  
15 35 256 1024 348,5

تمرين 2.2 - غيّر إلى النظام العشري والثنائي الأعداد السادس عشرية التالية :  
ABC 1A3B FFF 3A

تمرين 3.2 - إخصب المكمل إلى 2 للعدد 1A3B . أطرح 1A3B من العدد 2ABC .  
أعط التمثيل الموسّع إلى 32 بت للعدد 1A3B وكذلك لكملّه إلى 2 .

تمرين 4.2 - أعط القيم الرقمية العشرية التي تقوم بتأويلها :

C1F0000

- كمعطى ممثّل بفاصلة ثابتة حسب تكويد الاشارة والقيمة المطلقة .

- كعدد ممثّل بالمكمل إلى 2 .

- كعدد بفاصلة متحركة بطول قصير ( هل هو معايير في هذه الحالة؟ ) .

هل بالإمكان اعتبار هذا التشكيل الثنائي كمعطى مكوّد بالشكل العشري ؟

ما هو نقيض ( أو ضد ) هذا العدد في كلّ من التمثيلات المذكورة ؟

تمرين 5.2 - عاير العدد بفاصلة متحركة CS032000 .

### 3 . العنوان المطلقة ، العنوان النسبية

#### 1.3 . مميزات

في الفصل الأول عرضنا التعليمات - الآلية وكأنها مشكّلة من حقلين : الحقل كود العملية (operation code) وحقل العنوان . تحتوي التعليمة إذاً على العنوان المطلق للمتأثر ، أي عنوانه الفعلي أو الحقيقي بالنسبة للعنوان 0 من الذاكرة . هكذا في برنامج جمع مضمون الخلايا 0 و1 وتخزن النتيجة في العنوان 2 كان قد كتب على الشكل التالي:-

0		المتأثر الأول
1		المتأثر الثاني
2		النتيجة
3	8 8 0	$(0) \rightarrow AQ$
4	9 0 1	$AQ+1 \rightarrow AQ$
5	8 0 2	$AQ \rightarrow (2)$

فلنفترض بأننا زرنا هذا البرنامج ( مجموعة مناطق العمل والتعليمات ) ليس على العنوان 0 ولكن على العنوان 100 . سنكتب عند ذلك :

100		المتأثر الأول
101		المتأثر الثاني
102		النتيجة
103	8 8 1 0 0	$(100) \rightarrow AQ$
104	9 0 1 0 1	$AQ+(101) \rightarrow AQ$
105	8 0 1 0 2	$AQ \rightarrow (102)$

نلاحظ أن كود العمليات لا يتغير ولكن العناوين قد جرى نقلها 100 موقع لأن التعليمات تعود إلى العناوين المطلقة . أو بشكل آخر ، فإن كتابة البرنامج تتعلق بالعنوان الفعلي لكان البرنامج . هذا الإلزام ، الذي سنعرض سيناته ، قد أجبر مُصممي المكونات على تعريف أوالية العنوان النسبية : حقل العنوان من التعليمات لا يعود إلى العنوان المطلق للمتأثر ولكن إلى عنوان نسبي حسب عنوان أساسي (مطلق) . وبالإجمال فإن حقل العنوان يعطي « المسافة » إلى موقع المتأثر بالنسبة إلى عنوان يُعتبر وكأنه أساس أو قاعدة (Base adresse) ويعرف في لحظة زرع البرنامج في الذاكرة . العنوان الفعلي (المطلق) للمتأثر سيحسب ، في لحظة تنفيذ التعليمات ، بواسطة جمع العنوان المرجعي (الأساسي) إلى قيمة الإزاحة المحددة في التعليمات .  
ستعتمد في ما يلي إلى شرح أواليات عدة للعنوانات تتواجد في نفس الوقت على الآلات الحالية .

### 2.3 . العنوان القاعدية

هي عنوان نسبية حيث المبدأ هو كما ورد أعلاه . يحتوي الحاسب على عدد من المراسف التي يمكن أن تستعمل كمراسف أساسية (قاعدية) ، ويجب على المبرمج :  
- أن يختار المراسف الأساسي بواسطة أمر خاص .  
- أن يُخزن قيمة معينة في هذا المراسف ، قيمة ستكون عبارة عن العنوان الأساسي .  
- كتابة البرنامج (معطيات وتعليمات) نسبة إلى عنوان معين يعادل عادة الصفر .  
وفي لحظة التنفيذ يُسجن البرنامج في الذاكرة ، ويُخزن قيمة العنوان القاعدي في المراسف القاعدي . عند تنفيذ كل تعليمات فإن العنوان الموجود في التعليمات (الإزاحة (déplacement) يُضاف أوتوماتيكياً إلى مضمون المراسف القاعدي للحصول على العنوان الفعلي للمتأثر .

ذاكرة

1 5 0
-------

المراسف القاعدي

150		المتأثر الأول
151		المتأثر الثاني
152		النتيجة
153	8 8	0
154	9 0	1
155	8 0	2
156		

يُكتب البرنامج دون الإهتمام بالعنوان الفعلي لكان خزن البرنامج . ونحسب جميع العناوين نسبة إلى العنوان صفر (بداية البرنامج) .

ولنفترض إن بداية البرنامج ( العنوان النسبي صفر ) موجودة على العنوان الفعلي 150 ، وهي قيمة سيتم تخزينها في مصرف القاعدة<sup>(1)</sup> . إذا فالعنوان النسبي « للبرنامج يناسب العنوان الفعلي »  $n + 150 \dots$  والبرنامج سيقوم بتنفيذ العملية :  
 $(152) \rightarrow (151) + (150)$

لدينا إذن العلاقة التالية :

العنوان الفعلي = العنوان القاعدي + العنوان الموجود في التعليمات

نشير إلى أن عملية الجمع تتم ديناميكياً ، في لحظة تنفيذ كل تعليمات . يبدو من البديهي أن المبرمج لا يجب أن يُعدل مضمون المصرف القاعدي . العنوان النسبي الموجود في التعليمات يُدعى إزاحة (déplacement) .

المكتبات IBM 360/370 تتمتع بـ 16 مصرفاً عاماً يمكن أن تُستعمل كمراصف قاعدية . يُعَدُّ المصرف بالكامل بواسطة رقم المصرف المستعمل كمصرف قاعدي والعنوان النسبي . هكذا ، فإن حقل العنوان من تعليمات هذه المكتبات سيحتوي على حيز من أربع بتات حيث يتم تخزين رقم مصرف القاعدة .  
 الحسّنات :

- يكتب المبرمج برنامجاً بشكل مستقل عن الموقع الذي سيشغله في داخل الذاكرة .
- البرنامج ، أو مجموعة الحيزات والتعليمات ، هو قابل للتحويل والنقل . من الممكن نقله من حيز من الذاكرة إلى حيز آخر دون تعديل في العناوين المنقولة ( المحوّل ) . يكفي تعديل مضمون المصرف القاعدي .
- العنوان الأساسية وبشكل عام العنوان النسبية تسمح بعنونة مناطق كبيرة من الذاكرة بدلاً من أن تحتوي التعليمات ، على حقل عنوان طويل جداً . نشير حول هذا الموضوع ، أنه لعنونة  $2^n$  خلية من الذاكرة يلزمنا عدد  $n$  من البتات .  
 البتات :
- كل تعديل في مصرف القاعدة خلال تنفيذ التعليمات يؤدي إلى نتائج غير متوقعة .

### 3.3 ' العنوان المؤشرة (Indexed address)

يتعلّق ذلك بعملية حسابة العنوان بشكل شبيه بالعنونة القاعدية ولكن بهدف مختلف . يوجد مصرف يدعى مصرف التأشير أو مصرف الدليل (index register) ،

(1) العنوان القاعدي ليس بالضرورة عنوان زرع البرنامج .

يُخزَّن فيه قيمة معينة بواسطة المبرمج :  
هكذا :

العنوان الفعلي = العنوان القاعدي + الإزاحة + مضمون المرفص المؤشر

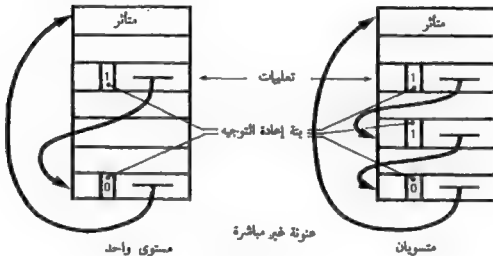
وعلى عكس مرفص القاعدة ، فإن مرفص التأثير يُمكن أن يُعدَّل مضمونه بواسطة المبرمج . هذه الأوالية تسمح ، بواسطة عمليات الزيادة على مضمونه هذاء بأن تقوم بعمليات تكرارية ، وتشكيل حلقات (loop) من التعليمات ، وبالتالي بلوغ خلايا متتالية من الذاكرة . هذه هي التقنية المستعملة لبلوغ الجداول . التعليمات التي تعود إلى عناوين والتي يُمكن أن تحتل عملية تأثير تتمتع بحقل إضافي خاص بالمرفص المؤشر حيث يستطيع المبرمج وضع رقم المرفص الذي يرغب باستعماله كدليل أو كمؤشر (index) .

#### 4.3 . العنونة المباشرة

نتكلم عن العنونة المباشرة. عندما نجد في التعليمة العنوان الفعلي للمتأثر . إنَّها إذن أوالية العنونة البسيطة والمطلقة .

#### 5.3 . العنونة غير المباشرة

هذه التقنية في العنونة موجودة على أكثر المكونات-حقل العنوان من التعليمة لا يحتوي على عنوان المتأثر ولكن على كلمة تحتوي عنوان المتأثر . بعض المكونات تتمتع ، عتادياً ، بأداة خاصّة لتغيير الإنهاء . في هذه الحالة ، يوجد بة خاصة في التعليمة تشير إلى وجود أو عدم وجود إعادة تغيير في الإنهاء . إعادة التوجيه يمكن أن تتم في مستويات عديدة كما يبرهن لنا المثل التالي :



### 6.3 . العنوان التلقائية

هذا المصطلح الشائع هو سيء لأن هذه الطريقة لا تخص عنواناً معيناً إنما تخص قيمة عيّنة . المعلومة الموجودة في حقل التعليمة المستعمل لكتابة العنوان ، لا تمثل عنوان المتأثر ، إنما المتأثر نفسه ( قيمة تستعملها التعليمة ) .

تصغير المرصف يمكن أن يتم بطريقتين :

- بواسطة العنوان المباشرة يتم تصغير كلمة من الذاكرة بعنوان A ، وسنستعمل تعليمة لشحن المرصف بعنوان مباشر مع مضمون  $R:A \rightarrow (A)$  ؛

- بواسطة العنوان التلقائية ، سيجري نقل القيمة صفر الموجودة في التعليمة على موقع العنوان إلى المرصف مع احتيال إزاحة البتة ذات الوزن الأكبر إلى اليسار إذا كان حجم حقل العنوان أصغر من حجم المرصف . العملية تتم بدون مساعدة أية خلية إضافية من الذاكرة . الحاسبات IBM 360/370 تتمتع بمجموعة من التعليقات ، تلك ذات الصيغة SI ، وتعمل بعنوان تلقائية .

## 4 هيكلية الحاسبات IBM 360 / 370

لن نقوم هنا سوى بإيجاز المميزات الضرورية الواجب معرفتها للبرمجة . بعض النقاط يمكن أن تعتبر حاجزاً أمام القارئ المبتدئ ، ومستوضح له لاحقاً إلا أننا وجدنا من المفيد تحديدها منذ الآن .

### 1.4 . الذاكرة

الذاكرة هي معنونة بالبايتات ( فقرة 1.5.2 ) . وسعتها القصوى هي 16777216 بايتة (2<sup>24</sup>) . تُرقم البايتات على التوالي بدءاً من الصفر . تجري التعليمات على سلاسل من البايتات ، نصف كلمات ( عناوين مزدوجة ) من بايتين ، وكل كلمات ( عناوين تقبل القسمة على 4 ) من أربع بايتات وكلمات مزدوجة ( عناوين مضاعفة لـ 8 ) من ثمان بايتات . تُرقم بتات الكلمات من اليسار إلى اليمين من 0 إلى 31 .

### 2.4 . المرافف

تستعمل مرافف التحكم بواسطة نظام التشغيل لإدارة الذاكرة . وهي مملوغة بواسطة تعليمات مميزة وخاصة ، لن نتكلم عنها .

المرافف العامة وعددها 16 ومُرَقمة من 0 إلى 15 ، ويمكن أن تُستعمل :  
- كمرافف قاعدية ( أساسية ) ( ما عدا المرصف 0 ) ، وتحتوي على عنوان مطلق من 24 بتة من اليمين .

- مرافف دلالية ( مرصف مؤشر ) ( index register ) ( ما عدا المرصف رقم 0 ) .  
- مرصف شحن ( مركم ) أو توسيع لمرصف الشحن يستعمل لإجراء العمليات على التمثيلات الداخلية للأعداد بفاصلة ثابتة أو عمليات منطقية . بعض العمليات تحتاج إلى وجود مرصفين « متلاحقين » ( ألضرب مثلاً ) . نستعمل عندئذٍ مرافف عامة متتالية ، الأول يكون إلزامياً برقم مزدوج . سنُسَمِّي لاحقاً زوجاً من المرافف كهذا ، مرصفاً مزدوجاً . التعليمات التي تستعمل مرصفاً مزدوجاً لا تشير سوى إلى المرصف برقم مزدوج .

المرافف الأربعة المتحركة هي متخصصة في الحسابات الجارية على الأعداد المثلثة بفاصلة متحركة . وتحمل الأرقام 0 ، 2 ، 4 ، 6 .

هذه المرافف هي بطول 64 بتة ويمكن أن تحتوي على عدد طويل بفاصلة متحركة أو عدد بطول قصير من نفس النوع . يشغل العدد القصير بفاصلة متحركة البتات ذات الأوزان العالية ، وتحمل البتات الأخرى . والمرافف المستعملة لتخزين الأعداد المثلثة بفاصلة متحركة أو المرافف المتحركة يمكن أن تزواج ( 2-0 و 4-6 ) بالنسبة للعمليات بالنسق الواسع (extended format) .

#### 3.4 . الكلمة PSW (Program status word)

الكلمة PSW هي عبارة عن كلمة مزدوجة متعلدة الأدوار . نجد فيها ، عند الانقطاع ، عنوان التعليمة التالية المطلوب تنفيذها . وتحتوي على نتائج عمليات المقارنة ( كود - الشرط ) ، ومعلومات عن بعض الحوادث ( كود الانقطاع ) . وتسمح بفتح حوادث الزيادة عن السعة (overflow) ، وتشير الى طريقة تشغيل الحاسب ( الصيغة الرئيسية أو المميّزة أو صيغة المسألة ) .

معرفة الكلمة PSW المرتبطة بالبرنامج تترجم إذا مفهومها الخاص بالتنفيذ . عند حدوث إنقطاع في البرنامج ، أي تعليق تنفيذه لمعالجة مسألة أكثر أولوية ، يتم تخزين الكلمة PSW الخاصة بالبرنامج المعلق في الذاكرة ، وتبقى عند ذلك الكلمة PSW القديمة . الكلمة PSW الجديدة ، والمرتبطة بالبرنامج الجديد الذي يعالج الانقطاع ، يتم شحنها مما يؤدي إلى تنفيذ برنامج جديد . البرنامج المعلق يمكن أن يعاود تنفيذه بشرط ترميم أي إستعادة الكلمة PSW .

هناك طريقتان للتحكم موجودتان على للكنة 370 : الطريقة الأولى (Basic control BC) والطريقة EC (Extended control mode) .

وتختلف الطريقتان من حيث كون الترجمة الديناميكية للعنوان هي غير ممكنة سوى في الطريقة EC . وبكل طريقة في التحكم نسق جديد للكلمة PSW . وتمييزها بواسطة البتة رقم 12 .

#### 1.3.4 . الكلمة PSW في الطريقة BC (bit 12= 0)

هذا هو نسق الكلمة PSW على للكنات IBM 360 .



خط 1.4 . نسق الكلمة PSW في الصيغة BC

- الأتمة . وهي مرتبطة بمختلف أسباب الانقطاعات . وجود البتة «0» في بتة القناع يمنع المعالجة المباشرة للحادثة . الإنقطاعات من نوع overflow (قناع البرنامج) يمكن أن تعمل ، وتوضع الأخرى في الانتظار حتى رفع أو زوال سبب المنع أو الإهمال . فقط بإمكان المبرمج بلوغ قناع البرنامج عندما يعمل الأخير في صيغة المسألة (15=bit) البتة رقم 15 تعادل 1 . .

البتات من 0 إلى 6 تتعلق بالإنقطاعات الآتية من القنوات . البتة 7 (E) ، الانقطاعات الخارجية ، البتة 13 (M) ، عمل المكنة السيء والبتات من 36 إلى 39 ، الانقطاعات الناتجة عن تجاوز في السعة ، البتة 36 مرتبطة بالفيض عن السعة (Overflow) أثناء إجراء العمليات الجبرية بفاصلة ثابتة ، والبتة 37 متعلقة بالنظام العشري والبتان 38 و39 متعلقتان بالحساب بفاصلة متحركة .

- مفتاح الحماية : هذا المؤشر (البتات من 8 إلى 11) ، وبالعلاقة مع المفتاح الموجود في الذاكرة ، يتيح أو يمنع بلوغ البرنامج إلى بعض المناطق من الذاكرة . البتة 12 (C) تشير إلى طريقة العمل في التحكم.  $C=0$  تدل على طريقة العمل BC . البتة 14 (W) ، تساوي 1 عندما تكون الوحدة المركزية غير فعالة ، في حالة الإنتظار (Wait) .

- البتة 15 (P) تعادل 1 عندما تكون الوحدة المركزية في الصيغة مسألة ، والتعليقات المميزة هي أيضاً ممنوعة . وتعادل هذه البتة صفراً في صيغة العمل (Supervisor) أي للمشرف .

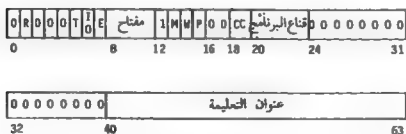
- كود الإنقطاع : عندما يحدث أي إنقطاع ، فإن الكلمة القديمة PSW للبرنامج المقطوع تُخزّن في الذاكرة ويوجد فيها كود خاص يُعرّف عن طبيعة الإنقطاع . - ILC (البتان 32 و33) (Instruction Length code) . عند حدوث إنقطاع نجد في هاتين البتتين طول آخر تعليمة جرى تفسيرها .

- CC (البتان 34 و35) . عبارة عن الكود - الشرط الذي يعطي نتيجة المقارنة ، إشارة المتأثر بعد تعليقات عديدة . . .

- عنوان التعليمة (البتات من 40 إلى 63) . هو عبارة عن عنوان التعليمة التالية المطلوب تنفيذها . تعرف PSW في لحظة الإنقطاع ، هذا الحقل يدل إذن على عنوان التعليمة حيث يجب أن يُعاود البرنامج عمله .

2.3.4 . الكلمة PSW في صيغة العمل EC (البتة 12=1) .  
تختلف عن السابقة بواسطة إلغاء أقمعة القنوات ، وكود الإنقطاع والكود ILC ، ويستبدل ذلك بواسطة قناع «RE» يدعى «program event recording mask» وبتة T

تتعلق بطريقة نقل العنوانين . دراسة هذه الإمكانيات تخرج عن إطار هذا الكتاب ، ولن نتكلم عنها .



شكل 4.2 . النسخ PSW في الصيغة EC

## 5 لغة الآلة

### 1.5 . نسق التعليمات الآلية

لقد آتت بنا دراسة المكنة البسيطة إلى تعريف التعليمات الآلية بطول ثابت ، والمركبة من كود للعملية ومن حقل للعنوان . تهتم العملية بتأثير واحد ، بينما يكون التأثير الثاني موجوداً في مصرف الشحن أو المرمك (Accumulator) .

تتمتع المكنات IBM 360/370 بأولية للعنونة أكثر تعقيداً ، تستعمل عدة مراصف وتتمتع بـ 16 مصرفاً عاماً يمكن أن تستعمل كمراصف شحن . نرى إذاً أن تعليمة بعنوان واحد ستكون مركبة من :

- كود للعملية (op. code) .
- رقم مصرف الشحن المعتمد في التعليمة .
- القسم عنوان الذي يتألف من :
- رقم المرفص القاعدي ،
- رقم المرفص الدليلي (للمؤشر) إذا كان مستعملًا ،
- قيمة الإزاحة .

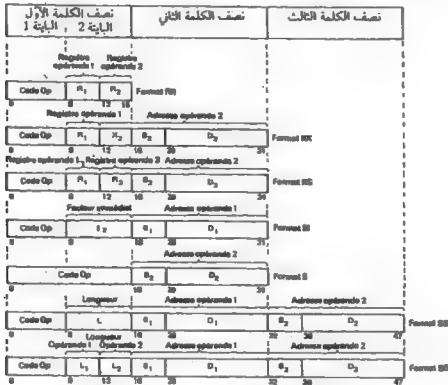
سيتم شرح تعليمات المكنات IBM 360/370 بواسطة ستة أشكال (نسق) مختلفة تتعلق بطبيعة التأثيرات . التعليمات ذات النسق RR (Register to Register) لا تستعمل سوى مصرفين . التعليمات من نوع RX تعالج عدداً موجوداً في أحد المرافص وآخر على عنوان معين في الذاكرة وهذا العنوان يمكن أن يكون دليلاً أو مؤشراً . النسق RS (Register and Storage) ، و SI (Storage Immediate) ، و SS (Storage and Storage) ، و S لا تسمح بأي عملية تأثير.

الجدول التالي يُجَدِّد نسق التعليمات المستعمل . الحقول R ، X ، B ، D تمثل على التوالي أرقام المرافص ، المرافص الدليلية ، مراصف القاعدة وقيمة الإزاحة . الحرف L يرمز إلى طول التأثير ويُقاس بالبايتة في التعليمات بالنسق SS . الدليلان 1 و 2

يرمضان هذه المعلومات بالمتأثر الأول والثاني .

ملاحظة إن البايته الأولى تحتوي دائماً على كود العملية ( ما عدا بالنسبة للنسق S الذي يستعمل 2 بايته ) ، إن نصفي الكلمة الثاني والثالث هما عبارة عن عناوين بشكل قاعدة وإزاحة . من المهم أن نتذكر أن التعليقات يجب أن تكون محصورة في نصف كلمات .

تتمتع التعليمة من نوع RX التي تستعمل عنواناً غير مؤشر بحيز X2 يعادل الصفر . والتعليمة التي تستعمل عناوين غير مرتكزة على قاعدة سيكون فيها الحيز II صفرأ . وبالتالي : فإن المرصف 0 لا يُستعمل لا كدليل ولا كمرصف قاعدي .



### جدول 1.5

كود العملية المتبني بـ

الطول بالبايتات

Code opération commençant par	النسق Format	الطول en octets
00	RR	2
01	RX	4
10	RX أو S , SI , RS	4
11	SS	6

### جدول 2.5

وفي النهاية ، يُمكن أن نُذكر بأن البتين رقم 1 و2 من كود العملية ترمزان إلى طول ونسق التعليمة . الجدول 2.5 يوجز لنا ذلك .

## 2.5 . فئات التعليقات

من الممكن تصنيف التعليقات الآلية ضمن ست فئات :

### 1 - تعليقات التبادل :

- من مصرف إلى مصرف .
- من الذاكرة إلى مصرف ( شحن المصرف LOAD ) .
- من مصرف إلى الذاكرة ( STORE ) .
- من الذاكرة إلى الذاكرة .
- شحن تلقائي لأحد المصارف .
- شحن تلقائي للذاكرة .

### 2 - التعليقات الحسابية :

- الجارية على أعداد بالنظام الثنائي البحث ( فاصلة ثابتة ) ،
- على أعداد بفاصلة متحركة ، بدقة بسيطة ، بدقة مزدوجة أو بنسق موسّع ،
- على أعداد بالنظام العشري المُكثف ،
- عمليات المقارنة الحسابية .

### 3 - التعليقات المنطقية :

- التقاطع ، الاتحاد ، للمكاملة ...
- المقارنة المنطقية .

### 4 - تعليقات التحكم بتوالي التعليقات ( تعديل مضمون عداد البرنامج PC ) .

- تفريع إلزامي . .
- تفريع مشروط .

### 5 - تعليقات الإدخال / الإخراج ( Input / Output )

### 6 - تعليقات متفرقة :

- تحويل النسق ، إختيار PSW ، الإزاحة ...

هذه التعليقات تعالج كلمات ، نصف كلمات ، كلمات مزدوجة أو سلاسل من السيات . إضافة لذلك نجد عدة تعليقات للجمع حسب طول المتأثرات ، ومواقعها في الذاكرة أو في المصارف ، أو حسب تكريرها الداخلي . مجموع التعليقات يتجاوز إذاً 150 تعليمة .

## 3.5 . كتابة البرنامج بلغة الآلة

هدف هذا المثل هو الإعتياد على نسق التعليقات الآلية . نقترح جمع مضمون

كلمتين وخزن النتيجة في الذاكرة .

كما ذكرنا أعلاه ، فإن جميع العناوين تُحسب بالنسبة إلى قاعدة ( أساس ) . الميم الأول للمبرمج هو في حفظ واحد من 15 مرصفاً علماً كمرصف قاعدي . نختار مثلاً المرصف رقم 15 .

هكذا ، فإن جميع التعليقات التي تستعمل عناوين مستحتوي عل «F» في الحقل المحفوظ للقاعدة .

كتابة البرنامج بلغة الآلة يتطلب اختياراً جيداً لعناوين وجود أو إدخال المعلومات في الذاكرة والمناطق المؤقتة لحفظ النتائج .

تسمح لنا أولية العنونة القاعدية والإزاحة بعدم الاهتمام بالعنوان الفعلي للمعلومات في الذاكرة . نعتمد في تفكيرنا العناوين النسبية . لنفترض إذا أن المتأثر الأول موجود على العنوان 0 والثاني في الكلمة التالية ، أي بدءاً من البايته رقم 4 . لنختر الكلمة الثالثة لتخزين النتيجة . ولنفترض أيضاً أن المتأثر الأول يعادل 29 والثاني يعادل 3 . فلنجعل حيز النتيجة صفراً في البداية . وكبي نستطيع تمثيل مضمون حيزات الذاكرة يجب علينا أيضاً تحديد طريقة التمثيل المعتمدة للأعداد . ولنختر الأسهل ، صيغة الأعداد بفاصلة ثابتة . حيز المعطيات في برنامجنا هو إذا مثل بالنظام السادس عشري على الطريقة التالية قبل تنفيذ البرنامج :

المتأثر الأول				المتأثر الثاني				المتأثر الثالث			
0	0	0	0	0	0	0	1	0	F	F	F
0	0	0	0	0	0	0	0	0	0	0	0
0				4				8			12

من الممكن تصوّر ثلاثة حلول مختلفة لكتابة برنامجنا :

### الحل الأول

شحن (LOAD) المتأثر الأول في مرصف نعتبره لاحقاً مرصفاً للشحن من نوع Accumulator ( يتم ذلك بواسطة تعليمة من نوع RX بين المرصف والذاكرة ) ، جمع المتأثر الثاني إلى هذا المرصف ( تعليمة RX ) ، وخزن مضمون المرصف في حيز النتائج ( تعليمة من نوع RX ) .

لنختر المرصف 2 كمرصف للشحن ( مركم ) . كود عملية تعليمة الشحن ( أنظر الملحق ) هو 58 ، والتعليمة تكتب بالنظام السادس عشري :

- حيز كود العملية (COP) 58
- الحيز R1 2 ( مرصف الشحن )

- حيز الدليل (index) 0 (بدون تأثير)
- مصرف القاعدة F (المصرف 15)
- الإزاحة 0

أي :

5	8	2	0	F	0	0	0
COP	R <sub>1</sub>	X <sub>2</sub>	B <sub>2</sub>		D <sub>2</sub>		

تمثل المعطيات بفاصلة ثابتة ، سنستعمل التعليمه بكود العملية 5A التي تؤمن جمع مضمون الخلية ذات العنوان B<sub>1</sub> + X<sub>2</sub> + D<sub>2</sub> إلى المصرف المذكور في الحيز R<sub>1</sub> أي :

5	A	2	0	F	0	0	4
COP	R <sub>1</sub>	X <sub>2</sub>	B <sub>2</sub>		D <sub>2</sub>		

004 = إزاحة التأثير الثاني بالنسبة إلى القاعدة .

وفي النهاية ، سنخزن النتيجة ( التعليمه STORE ، بالكود 50 ) في الكلمة الثالثة على العنوان 8 .

5	0	2	0	F	0	0	8
COP	R <sub>1</sub>	X <sub>2</sub>	B <sub>2</sub>		D <sub>2</sub>		

بإمكاننا أن نفحص صورة البرنامج بعد تخزينه في الذاكرة .

العناوين الموجودة هنا هي العناوين النسبية ولا تتأثر بالعنوان الفعلي لموقع تخزين البرنامج . عنوان الإطلاق في التنفيذ ، أي عنوان أول تعليمه لتنفيذ ، هو عنوان القاعدة C + .

0	0	0	0	0	0	0	1	D
4	F	F	F	F	F	F	F	D
8	0	0	0	0	0	0	0	0
C	5	8	2	0	F	0	0	0
10	5	A	2	0	F	0	0	4
14	5	0	2	0	F	0	0	8
18								

عنوان الإطلاق  
في التنفيذ

### الحل الثاني :

أشحن المتأثرين الأول والثاني في المرصف ، وقم بعملية جمع لمضمون مرصف مع المرصف الآخر ومن ثم خزّن النتيجة . نستعمل المرصف 2 و 3 كمرصف للعمل والمرصف رقم 15 كمرصف قاعدي . والبرنامج هو التالي :

0	0	0	0	0	0	1	0	
4	F	F	F	F	F	F	F	0
8	0	0	0	0	0	0	0	0
C	5	8	2	0	F	0	0	0
10	5	8	3	0	F	0	0	4
14	1	A	2	3				
16	5	0	2	0	F	0	0	8

شحن المتأثر الأول في R2  
شحن للمتأثر الثاني في R2  
جمع في R2  
خزن النتيجة

هذا الحل يحتاج إلى تعليمة إضافية . ستلاحظ وجود تعليمة من نوع RR بطول 2

بأيتة .

### الحل الثالث :

الحل الثالث كان سيقوم على إجراء الحساب مباشرة في الذاكرة دون استعمال المرصف . وسيتحتاج إلى وجود تعليمة بثلاثة عناوين ( المتأثر الأول ، للمتأثر الثاني والنتيجة ) . إلا أن هذا النوع من التعليمات هو غير موجود هنا .

### خلاصة

نلاحظ ، في الأمثلة المذكورة ، أن حيز المؤشر (index zone) غير المستعمل هو مصفّر تماماً كما ذكرنا في الفقرة 1.5 .

إن البرمجة بلغة الآلة تبدو معقّلة ودقيقة رغم بساطة المثل وعدم إتمامه . لهذا السبب لا نستعمل هذا النوع من البرمجة ونفضّل عليه مرونة لغة المؤلّ (الأسمبلر) .

## 6 . لغة المؤول

### ASSEMBLER

المثل البسيط الذي جرى عرضه في الفصل السابق أثبت لنا جميع صعوبات البرمجة بلغة الآلة مع أنه جرى تبسيط كبير لعملنا باستعمال النظام السادس عشري بدلاً من النظام الثنائي .

في لغة الآلة ، فإن أكواد العمليات والعناوين هي رقمية . وكل تعديل في موقع المعطيات يؤدي إلى تعديل العناوين في التعليقات المتعلقة بها .  
هذه الصعوبات أدت بالمصممين إلى تعريف لغة ، تُدعى المؤول (assembler) ، قريبة من لغة الآلة ولكنها سهلة الإستعمال مما يجعل ترتيبها في مصاف اللغات المتطورة .

#### 1.6 . مميزات لغات التأويل

- 1- تتميز التعليقات بلغة المؤول بكود عمليات تذكيري . مثلاً : تعليمة شحن المرصف من خلال مرصف آخر تتمتع بكود رمزي هو LR (LOAD TYPE RR) ، وتمتاز تعليقات الجمع بكود رمزي يبدأ بالحرف A ...
- 2- بإمكان المبرمج أن يقوم بتحديد عناوين بواسطة أسماء رمزية ويقوم برنامج ترجمة المؤول إلى لغة الآلة بربط القيمة الرقمية المناسبة بهذه الأسماء .
- 3- تتمتع لغة المؤول ليس فقط بمجموعة التعليقات الآلية التي تتضمنها لغة الآلة ، ولكن ببعض التعليقات الخاصة الآلية التي تدعى (التوجيهات) (أو أشباه التعليقات Pseudo-Instruction) وبعض الماكرو تعليمات (macro-instructions) .

#### 2.6 . تعريفات

تدعى تعليمة - آية كل تعليمة مكتوبة بلغة المؤول ومترجمة إلى تعليمة واحدة فقط بلغة الآلة . يتناسب كود رقمي مع كود - العملية التذكيري . مثلاً ، عملية نسخ المرصف 12 في المرصف 3 ، تكتب بلغة المؤول حل الشكل التالي :

وتُترجم إلى لغة الآلة بواسطة :

1	8	3	C
OP	R <sub>1</sub>	R <sub>2</sub>	

يُدعى أمر من نوع توجيه directive كل طلب إلى المؤول ، لا يُؤلّد أبداً تعليمة آلية ولكن يُقدّم توجيهات للتأويل والتجميع . يوجد نوعان من التوجيهات : تلك التي لا تؤدي إلى أية عملية حجز للذاكرة وتلك المستعملة لحفظ موقع من الذاكرة أو تعريف الثوابت المفيدة للمسألة . هكذا ، فالتعليمة USING\*,15 تعني إن المرفص 15 سيُعتبر أولاً كمرفص قاعدي ، مما سيسمح بعدم ذكر القاعدة (Base) في التعليمات التالية . هذا التوجيه لا يشغل مكاناً من الذاكرة في الكود المؤلّد ، وليس هو سوى إشارة إلى برنامج التأويل والتجميع . أن نكتب 'X'FOFO' DC يعني أن نطلب إلى المؤول حجز بايتين من أجل تخزين الثابتة المحددة بالنظام السادس عشري بواسطة FOFO . لا يوجد توليد لتعليمة ولكن فقط حفظ لمكان من الذاكرة . من الممكن تشبيه هذه التوجيهات بتعليقات التصريح في اللغات المتطورة . أن نكتب بلغة فورتران الأمر dimension TAB (100) يعني أن نطلب من المرفص (Compiler) حفظ المكان من الذاكرة اللازم لاستيعاب الجدول TAB (100) .

سنسمّي ماكرو - تعليمة (MACRO-INSTRUCTION) كل طلب إلى البرنامج المؤول assembler باستبدال سلسلة معلّفة مسبقاً من التعليمات تدعى ماكرو - تعريف . للماكرو تعريف هو إذاً عبارة عن مجموعة من التعليمات ينسخها البرنامج assembler مكان كل ماكرو - تعليمة . يقدم النظام مجموعة من الماكرو - تعريفات تدعى نموذجية (مستاندرد) تُسهّل على المبرمج القيام ببعض العمليات المعقّدة ، كعمليات الإدخال - الإخراج . كما باستطاعة المبرمج أن يقوم بتعريف نظام خاص به من الماكرو - تعريفات .

### 3.6 عملية التأويل

الاسم «assembler» يعني في نفس الوقت اللغة والبرنامج الذي يقوم بترجمة النص إلى لغة - الآلة . سنقوم هنا بتناول مرحلة الترجمة بصورة موجزة . يبدو المؤول وكأنه عبارة عن مرفص أو كأنه عبارة عن برنامج لترجمة النص المكتوب بلغة منبع إلى نص مستهدف يتألف من تعليمات - آلية . تدعى عملية الترجمة تأويلاً «assembling» .



### 1.3.6 . عداد المواقع

يجب على المؤول ، ومن خلال نص منبع ، أن ينتج نصاً ثنائياً يكون مع بعض التحويلات عبارة عن صورة البرنامج المطلوب تنفيذه . لتخصيص عناوين متتالية للتعليقات ، يستعمل المؤول عدداً للمواقع نرمز إليه بواسطة CE . في بداية عملية التاويل فإن CE يسوى ، مثلاً بصفر . وخلال ترجمة التعليقات فإنه يزيد من قيمته حسب طول التعليقات المترجمة . وعندما يلتقي توجيهاً من نوع حجز موقع أو منطقة من الذاكرة ، فإن مضمون CE يزداد حسب طول المنطقة المحجوزة . كل توجيه من نوع إشارة إلى برنامج التاويل لا يؤدي إلى زيادة في مضمون CE لعدم توليد أية تعليمة آلية . التعليقات ذات النسق RR تؤدي إلى زيادة مضمون CE 2 بايت ، أما تلك التي تتمتع بنسق RX ، RS و SI فتؤدي إلى زيادة أربع بايتات إلى مضمون CE ، أما تلك ذات النسق SS فتؤدي إلى زيادة 6 إلى مضمونه . وكل توجيه لحجز كلمتين من الذاكرة يؤدي إلى زيادة مضمونه 8 بايتات .

في المثل التالي ، STARTO هي عبارة عن توجيه يؤدي إلى تهيئة CE وتصغيره . لا يحدث أي توليد لتعليقات جديدة وبالتالي فإن CE يبقى صفراً . STM 14, 12, 12(13) هي عبارة عن تعليمة من نوع RS تؤدي إلى زيادة 4 إلى مضمون CE . والتوجيه DS 1F يؤدي إلى حفظ كلمة من الذاكرة يرمز إليها بواسطة ALPHA . CE تزداد قيمته 4 بايتات . التعليمة LR 0,1 بالنسق RR تجعل مضمون CE يزداد 2 .

ملاحظات	منطقة التكرار كود العملية	الرمز الرمزي	CE بالنظام السادس عشري
تصغير CE تعليمة من نوع RS	START 0 STM 14, 12, 12(13)		0 0 4
حجز كلمة	DS 1F	ALPHA	20 24
تعليمة من نوع RR	LR 0,1	DEBUT	48 4A

وبالاختصار ، فإن عدد المواقع هو عبارة عن كلمة - ذاكرة يُخزّن فيها المؤول :  
 قبل التأويل التعليمية ، عنوان بداية التعليمية ( المتعلّق بنهية CE ) ،  
 - بعد التأويل ، عنوان الخلية الأولى المتوفرة .  
 من الممكن أن نلاحظ إن قيمة CE تعادل قيمة مضمون عدد البرنامج عند  
 التنفيذ .

### 2.3.6 . العنونة الرمزية والمرجعيات المطلقة

لقد ذكرنا سابقاً أنّ أحد أهم مميزات وخصائص المؤول تكمن في إمكان تسمية  
 العناوين والقيم بواسطة رموز . يمكن أن يكون الرمز عبارة عن إسم منطقة من  
 الذاكرة . في الجدول السابق ، فإن ALPHA و DEBUT هما عبارة عن عناوين رمزيين  
 نستطيع بلوغهما والعودة إليهما . سيكون بإمكان المبرمج أن يراجع مناطق من الذاكرة تبعاً  
 لهذين العنوانين بواسطة تعابير من نوع  $2 + \text{ALPHA} - \text{DEBUT}$  .

يُستعمل الرمز \* لتسمية القيمة التي يأخذها CE في لحظة التأويل ، أي عنوان  
 البداية اليسرى من التعليمية الموجودة في طور التأويل . من الممكن أن نعود أيضاً بواسطة  
 2- \* إلى العنوان الجاري ناقص 2 بايت .

سنلاحظ أيضاً أنّه لا يمكن لقيمتين مختلفتين لمضمون CE أن يحملتا نفس  
 الإسم . إذ نكون عندئذ في حالة التعريف المزدوج .

يسمح المؤول أيضاً ببلوغ قيم مطلقة بواسطة رموز ، أي رموز غير متغيرة عند  
 ترجمة البرنامج . نكتب عملية نسخ المرصف 1 في المرصف 0 مثلاً :  $\text{LR } 0,1$  .

يمكننا أيضاً أن نكتب ، بشكل أوضح  $\text{LR } R0, R1$  بشرط تحديد كون  $R0$  و  $R1$   
 عبارة عن رمزين مطلقيين . يعادلان القيمتين 0 و 1 .

وفي النتيجة ، فإن للمؤول سيربط بكل رمز قيمة تدمي قيمة - خاصة ، وهذه  
 القيمة سيتم ترجمتها أو علمه حسب الحالة .

### 3.3.6 . جدول الرموز

عند العمل ، وفي كل مرة يلتقي المؤول رمزاً معيناً في منطقة الوسم (Label) يقوم  
 بتخصيص خاصيات له :

- خاصة - قيمة تعادل قيمة CE في هذا الموقع .
  - خاصة - طول تعادل البعد (الحجم) بالبايتات للمنطقة المينة .
- يمكن أن يقوم المؤول إذاً ببناء جدول من الرموز على الشكل التالي :

خاصية - طول	خاصية - قيمة	وسم رمزي
4	20	ALPHA
...	...	BETA
2	48	DEBUT
...	...	...

عندما يلتقي رمزاً معيناً في قسم العنوان من التعليمة ، يقوم المؤول بإستشارة هذا الجدول . فإذا كان هذا الرمز موجوداً فيه معناه ذلك أنّ الرمز محدد مسبقاً ، وإلا فذلك يعني مرجعاً إلى الأمام ، أي إنه لم يلتقي الرمز حتى الآن في منطقة الـ 20 و 21 . لاحقاً ( إلا إذا كان يتعلق ذلك برمز خارجي ، أنظر الفصلين 20 و 21 ) .

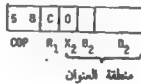
#### 4.3.6 . تأويل التعليمة

يتعلق ذلك باختيار كيفية ترجمة التعليمة بواسطة المؤول وبالأخص كيف يقوم بتحويل العنوان الرمزي إلى عنوان قاعدي ، مؤشر وإزاحة . سنقوم بتحليل ذلك من خلال مثل معين .

لنفترض التعليمة التالية :

L                      12, ALPHA  
المعامل      المعامل  
الثاني      الأول

إنّها تعليمة من نوع RX ويكود عملية 58 ( أنظر الملحق ) حيث معناها هو « شحن مضمون الخلية ذات العنوان ALPHA في المرصف رقم 12 » . يقوم عمل المؤول على تمثيل مختلف حقول التعليمة بالنسق RX ، أو :



فلنلاحظ منذ الآن إن منطقة الدليل هي صفر ، لأنه لم يذكر أي مرصف مؤشر أو دليل في المعامل الثاني من التعليمة ( الحقل الثاني منها ) . ولتكتملة حيز العنوان - يجب :

- معرفة المرصف المستعمل كقاعدة ،
- معرفة إزاحة العنوان ALPHA بالنسبة للعنوان القاعدي .

ونشير إلى أن العنوان القاعدي لا يختلط بالضرورة مع عنوان وجود البرنامج في الذاكرة .

سنقوم بافتراض في التل إن ALPHA تناسب القيمة 1C للعدد CE ، وإن المرفص 15 هو مرفص القاعدة وإن العنوان القاعدي يناسب القيمة C للعدد CE .  
إزاحة ALPHA بالنسبة إلى القاعدة هي إذا C-C1 أي 10 . التعليملة الآلية المؤولة ستكون إذا :

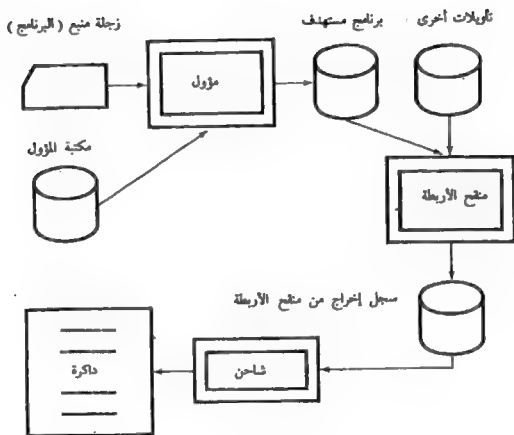
5	8	C	0	F	0	1	0
---	---	---	---	---	---	---	---

#### 4.6 . مراحل تنفيذ البرنامج

إن تنفيذ البرنامج المكتوب بلغة المؤول ، كما بالنسبة للبرنامج المكتوب بلحدي اللغات المتطورة ، يتطلب عدة مراحل . المرحلة الأولى هي مرحلة التأويل والتجميع التي تكلمنا عنها . يُترجم النص الأولي إلى لغة الآلة ويُسخ في سجل على الاسطوانة المغناطيسية . المرحلة الثانية ، التي يمكن أن تكون اختيارية للبرامج البسيطة ، هي تنقيح الأربطة (link editor) . وتؤدي إلى إجراء بعض الوصلات بين مختلف الزجل للمؤولة بشكل منفصل أو التي تشكل جزءاً من مكتبة البرامج . متفح الأربطة يُشكل زجلة واحدة مستهدفة ، يمكن أن تتمتع ببيكلية تغطية ، من خلال مختلف عمليات التأويل . المرحلة التالية تقوم على شحن الزجلة في الذاكرة ، أي إعطائها عنواناً فعلياً لحزنها . وفي هذه الحالة تكون العناوين القاعدية متجمدة ، وبعض المعلومات المتعلقة بالعناوين للطلقة يجب أن تُحسب من جديد . يكفي إذا أن نقوم بتخزين عنوان أول تعليملة للتنفيذ في عداد البرنامج CO ( للكلمة الثانية من PSW ) للبدء بمرحلة التنفيذ .

سنسمي نقطة الشحن أو عنوان الحزن ، عنوان بداية المنطقة المخصصة للبرنامج . سُدعى عنوان الإطلاق عنوان أول تعليملة للتنفيذ من البرنامج . نقاط الدخول إلى البرنامج هي عناوين ، التعليملات أو المعطيات ، من الممكن بلوغها من خارج البرنامج . تتصل نقاط الدخول هذه بمتفح الأربطة الذي يمكن أن يقوم بإجراء وصلات بين مختلف الزجل (modules) . عنوان الإطلاق هو نقطة دخول .

بدون إعطاء جميع الإمكانيات فإن المخطط 1.6 يمرض مختلف المراحل الواجب أن يتبعها البرنامج كي يجري تنفيذه .



شكل 1.6



## القسم الثاني

### المؤول 360/370

#### 7 . العناصر الأساسية

##### 1.7 . عموميات وتقديم البرنامج

###### 1 - مجموعة السيات :

يستعمل المؤول السيات الأبعدية A ، B ، Z... ، @ ، \$ ، والأرقام 0 ، 1 ، 2 ... 9 ، والسيات الخاصة : + - \* / = ( ) ، ' ، & والقسم البيضاء (blank) .

###### 2 - ورقة البرنامج

المنطقة المحجوزة للمؤول تمتد من العمود 1 إلى العمود 71 . المنطقة 73 إلى 80 لا تُفسر من جانب المؤول وتُستعمل لتحريف التعليقات . العمود 72 يُستعمل عندما ترغب إحدى التعليقات بالمتابعة على السطر التالي . تقسم منطقة التعليمة ( 1 إلى 71 ) إلى أربعة أقسام :

منطقة الرموز : وتُستعمل لأجراء تخصيص رمزي للتعليمة ( وسم ) أو إلى معطى ( اسم المعطى ) .

###### الاسم المُخصص :

- يبدأ بالعمود 1 بواسطة صمة أبجدية .
- يحتوي على أكثر من 8 سيات أبجدية .
- لا يحتوي على فراغ أو سيات خاصة .

الرموز التي تظهر في منطقة التأثيرات تخضع لنفس القواعد :

أمثلة :



## 2.7 . عناصر لغة المؤول

لقد لاحظنا حتى الآن إن المؤول يسمح لنا باستعمال رموز معينة لتسمية العناوين أو القيم . وعملياً فإن لغة المؤول تسمح لنا :

- باستعمال كتابات مثل 'B'1011' ، 'X'1A10C' والتي ستعامل وكأنها قيم باللغة الثنائية ، أو السلاسل عشرية . . . وهي ستكون عبارة عن القيم المعرفة أوتوماتيكياً .

- بلوغ الطول التعلق بأحد الرموز . لو افترضنا إن «BIDON» هو رسم تعليمية ، أو بشكل عام ، أكثر اسم حيز معين ، فإن L'BIDON سيحدد طول التعليمية أو المنطقة . ويتعلق ذلك بالخاصية - طول :

- استعمال الأحرف كمثابرات في التعليقات ؛

- خلط كل هذه الإمكانيات لنحصل على تعابير ستكون معادلة لعناوين قابلة للنقل إلى قيم مطلقة .

من اللازم إذاً تحديد القواعد النحوية التي تسمح باستعمال هذه الإمكانيات

### 1.2.7 . قيم المعرفة الأوتوماتيكية (Auto-definition)

قيمة المعرفة الأوتوماتيكية هي واحد من أشكال الكتابة ، معروف من قبل المؤول ، يسمح بتحديد القيمة .  
مثلاً :

'XB' ، 'B'1011' و 11 هي عبارة عن ثلاث كتابات مختلفة تسمح بتحديد القيمة 11 (عشري) المثلة في المكنة بواسطة تشكيلة البتات 1011 . هذا الشكل في الكتابة هو مسموح ، مع بعض التحديدات ، بداخل حيز العوامل (منطقة العنوان) من التعليمية .

هناك أربعة أنواع من المعرفة الأوتوماتيكية المقبولة :

- الثاني : 'B'1001101' وعلى الأكثر 32 رقماً ثنائياً تحت إشراف النظام OS و 24 بالنظام (DOS) .

- السادس عشري : 'X'1A3BC' ،

- العشري : 125 (حد أقصى 10 أرقام عشرية) .

- نوع السهات : 'C'A' ، 'C' (سمة أبوستروف أو الفاصلة العليا) ، 'C'ABCD' ، 'C'AB' . يجب أن نحصل كحد أقصى على أربع سهات بالنظام OS وثلاث بالنظام DOS .

وبشكل عام ، فإن قيمة التعريف الأوتوماتيكية يجب أن تتم على 24 بتة بإشراف النظام DOS وعلى 32 بتة كحد أقصى بإشراف النظام OS . سنجد أمثلة على طرق استعمالها في الفقرة 3.7 للمتعلقة بالتعابير .

## 2.2.7 . التأثيرات الحرفية

- هي عبارة عن قيم مستعملة كمثاثرات في حيز عوامل التعليمات .
- لشحن القيمة 125 في المرصف 3 يمكن للمبرمج أن يختار أحد حلين :
- 1- حيز حيز من الذاكرة ، يدعى ALPHA مثلاً ، ويُعرّف عنه وكأنه يحتوي على القيمة 125 ، وبعد ذلك يُشحن ALPHA في المرصف 3 بواسطة التعليمات :  
L3, ALPHA;
  - 2- كتابة التعليمات :  $F'125 = L3$  ، وسيتهم المؤول بسجّز الخلية من الذاكرة التي تحتوي على 125 في منطقة تدعى POOL (حوض) .  
في المثل المذكور لاحقاً ، فإن القارىء سيتحقّق :  
- من أن المؤول سيضع عنوان المثاث الحرفي بشكل قاعلة وإزاحة داخل كود التعليمات المؤلّد عنه ،  
- من أن إستعمالين مختلفين لنفس المثاث الحرفي لن يؤدّيا سوى إلى حيز واحد في الذاكرة ،  
- من أن المثاث الحرفي هو شبيه برمز قابل للترجمة .  
إنّ استعمال المثاث الحرفي ، إن لم يجعل أي شيء جديد ، فإنه يُقدم لنا فائدة بالنسبة لوضوح كتابة التعليمات .
- قواعد الكتابة
- يُعَدّ المثاث الحرفي وكأنه مثاثّر عادي في توجيه DC مسبوق بالإشارة « = » . أما القواعد المتعلقة بمثاثرات التوجيه DC فإنها ستوضح لاحقاً .
- لا يمكن أن يُستعمل المثاث الحرفي كمُعْادِل في التعبير (فقرة 3.7) الرقمي أو غير الرقمي .
- من البليبي ، لأن المثاث الحرفي يُستعمل « كمعطى للإدخال » في التعليمات ، أن لا يظهر في الحقل المُستقبل من التعليمات . سيكون من المتأفّر أن نكتب :  $ST3 = F'125$  (ST = خزّون مضمون المرصف في الذاكرة) .

L'7C	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
007000				1	CSECT
				2	EXTRN
				3	USING
007000	5810 F018	00019	00000	4	1=0*
007004	5820 F01C	0001C		5	2=0*ABCD*
007008	5820 F018	00018		6	3=0*
00700C	5820 F020	00020		7	4=V(SPI)
007010	5830 F01C	0001C		8	5=0*ABCD*
007014	5810 F024	00024		9	6=A(SPI)
				10	END
007018	00000000			11	7=0*
00701C	C1C2C3C4			12	8=C*ABCD*
007020	00000000			13	9=V(SPI)
007024	00000000			14	10=A(SPI)

### 3.2.7 . الخاصية - طول

وتسمح ببلوغ الطول المرتبط بالرمز . ويكتب :

مثلاً : L' symbolic name L' اسم رمزي

L'ZONE L'SUITE L'\*

- إذا كان الرمز هو اسم الحيز ، فهو يأخذ كقيمة طول الحيز مُقاساً بالبايتة .
- إذا كان الرمز هو اسم التعليمة ، فهو يأخذ واحدة من القيم 2 ، 4 أو 6 حسب نسق التعليمة .
- إذا كان الرمز هو \* ، فهو يأخذ كقيمة طول التعليمة التي يظهر فيها .
- بالنسبة للتوجيهين DC و DS . فإن الخاصية - طول لا تتأثر بوجود عامل الإزدواجية . ملاحظ أنه بالنسبة للتوجيه BQU فإن قيمة الخاصية - طول هي قيمة التأثير الأيسر .
- الأمثلة التالية ، والفهم الكامل ، تتطلب بأن نكون أكثر تقدماً في هذه الدراسة .

إلا أننا نمرضها هنا :

الرمز	كود - العملية	عوامل	خاصية	قيمة
ZONE1	DS	CL80	L'ZONE1	80
ZONE2	DS	CL200	L'ZONE2	200
CARAC	DC	C'ABCDE'	L'CARAC	5
ABSOL1	EQU	ZONE2-ZONE1	L'ABSOL1	200
ABSOL2	EQU	25	L'ABSOL2	1
INSTR1	LR	0,1	L'INSTR1	2
			L'*	2
INSTR2	MVC	ZONE2(L'*) , ZONE1	L'INSTR2	6
			L'*	6
	MVC	ZONE2(L'ZONE2-10) , ZONE1	L'ZONE2	200
ALPHA	DC	6F'0'	L'ALPHA	4

### 3.7 التعابير

تعريف :

التعبير هو تركيب من الرموز ، وقيم التعريف - الأوتوماتيكي وخصائص - الطول في منطقة التأثيرات من التعليمة .

الاستعمال :

تستعمل التعابير لتحديد :

- العنوان ،
- الطول الواضح ،
- المعدل ،
- عامل التكرار .
- التأثير .

نظمت التعابير

التعابير هي بسيطة أو مركبة ، مطلقة أو قابلة للترجمة . التعبير البسيط هو الرمز الوحيد أو الرمز « $\bullet$ » ( قيمة عدّد المواقع عند تأويل التعليمة ، فقرة 1.3.6 ) .  
التعبير المركّب هو مجموعة من عدة تعابير بسيطة مرتبطة بمؤثرات من نوع + ، - ، \* ، (1) أو / ، التي تمثّل على التوالي الجمع ، الطرح ، الضرب والقسمة .  
أمثلة :

ALPHA+2	++3	* =	CE
ALPHA-BETA	--2	* =	CE
3*DELTA	A*3	* =	مؤثر
(ALPHA-BETA)/2	++2		تعبير غير صالح
ALPHA*X'1A'	*3		تعبير غير صالح
TAB+L'LIGNE			

قواعد الإنشاء

التعبير المركّب :

- لا يمكن أن يبدأ بمؤثر ،
- لا يمكن أن يحتوي على مؤثرين ثنائيين متتاليين ،

(1) يجب عدم الخلط بين المؤثر \* والرمز الذي يمثل عدده المواقع .

- لا يمكن أن يحتوي على نجمتين ،
  - لا يمكن أن يحتوي على تعبيرين بسيطين يتابعان بلون مؤثر بينهما ،
  - لا يمكن أن يحتوي على متأثر حرفي .
- النظام OS يسمح باستعمال 19 مؤثراً أحادياً وثنائياً و6 مستويات من الأهلة . بينما النظام DOS لا يسمح سوى بـ 15 مؤثراً و5 مستويات .
- تقييم التعابير

يقوم المؤول بتخصيص قيمة رقمية لكل تعبير بسيط وبعد ذلك يُقِيم من اليسار إلى اليمين التعبير حسب أولوية خاصة للضرب وللقسمة بالنسبة للجمع والطرح .  
 $A+B * C$  تُقِيم وكأنها  $A+(B * C)$  وليس كأنها  $(A+B) * C$  . النتيجة الحسابية تصبح قيمة التعبير ، والمؤول يُقِيم بشكل طبيعي في المكان الأول المؤثرات الأحادية وداخل الأهلة . القسمة على صفر هي صبيحة وتعطي نتيجة صفر .

تعابير مطلقة ، تعابير منقولة

التعبير المنقول هو تعبير حيث القيمة تتغير مقدار n إذا كان البرنامج منقولاً إلى n بائنة

التعبير المطلق هو التعبير الذي لا تتغير قيمته عند النقل .  
 أمثلة :  
 لنفترض أن ALPHA و BETA هي رموز منقولة وإن VAL1 و VAL2 هي رموز مطلقة :

تعابير مطلقة	تعابير منقولة
VAL1+B*101	ALPHA+3
ALPHA-BETA	BETA+L*ZONE
VAL1+VAL2	BETA+VAL1

- التعبير سيكون مطلقاً إذا كان يحتوي على :
- رموز مطلقة ، قيم تعريفات أوتوماتيكية ، خاصيات - طول ،
  - رموز منقولة يظهر كل اثنين منها على حدة وتؤدي إلى تصغير فاعلية النقل .
- ملاحظة : إنه إذا كان T1 و T2 تعبيرين منقولين ، فإن T1+T2 و T1\*3 و T1 ليست لا مطلقة ولا منقولة .

ولتأكد من ذلك يكفي أن نقوم بإجراء عملية نقل 100 مثلاً :

T1 + 100	تصبح	T1
T2 + 100	تصبح	T2
T1 + T2 + 100	تصبح	T1 + T2
T1 * 3 + 300	تصبح	T1 * 3

التعابير لا تحتل نفس الإزاحة .

إستعمال التعابير هو بشكل خاص مفيد لأنه يسمح بتحليل العناصر حيث القيم هي قابلة للتغير عند التحويل وذلك بشكل مُعاملات ومتغيرات ( مثلاً صفحة 122، السطر 78 من البرنامج ) . كل تعديل في قيمة المتغير من التعبير سيكون محسوباً من جديد بواسطة المؤول وليس بواسطة المبرمج ، مما يُسهّل عمل المبرمج .

## 8 توجيهات تعريف الرموز

لنأخذ هذه القطعة من برنامج بلغة فورتران :

```

DIMENSION TAB(100)
DO 50 I=1,100
  TAB(I)=I
50

```

يطلب الأمر DIMENSION حجز 100 كلمة - ذاكرة مجموعة تحت إسم الجدول TAB . تدل القواعد الضمنية المتعلقة بنوع المرفقات أن هذا الجدول سيتألف من أعداد حقيقية ، أي مكونة في التمثيل بفاصلة متحركة بدقة بسيطة . يعرف المصرف بأنه يجب أن يستعمل ، لتوليد كود التعليقات الحسابية التي تبلغ TAB ، التعليقات الحسابية بدقة بسيطة .

وفي فورتران ، كما في جميع لغات البرمجة ، كل رجوع إلى معرف يفترض أن يكون الأخير معروفاً من المصرف ، أي محدداً خلال البرنامج بواسطة نوعه ( حقيقي ، صحيح .. ) وطوله مقاساً بالكلمات أو بالبايتات . وفي النهاية يختص المرف TAB بخاصية - مهمة ( قيمة المرف ستكون عنوانه ) ، وبخاصية - طول ( بعد الحيز المشار إليه بالبايتة ) .

في لغة التاويل المسألة هي نفسها ، يجب أن يحدد كل رمز بواسطة خواصه . سنرى توجيهين DC و DS يسمحان بتعريف الثوابت وحجز مكان من الذاكرة ، والتوجيه EQU الذي يسمح بإجراء توازنات بين الرموز .

### 1.8 . تعريف الثابتة DC

كثير الإستعمال ، هذا التوجيه يسمح بحجز منطقة من الذاكرة تحتوي على القيمة المدعومة ثابتة وتسميتها بواسطة أحد الرموز .

شكل هذا التوجيه هو التالي :

عامل	كود العملية	رمز
$d t m \text{ } ^{\circ} c^{\circ}$	DC	[ وسم ] 1

- الوسم هو الإسم الرمزي للثابتة وهو اختياري .
- $d$  هو عامل الازدواجية ، وهو اختياري ، وإذا كان مهملًا فإن قيمته تعادل 1 . إنه يشير إلى العدد الذي يجب أن تولد فيه الثابتة .
- $t$  هو النوع ، يمكن أن يكون أحد الأكواد الموجودة في الجدول التالي :

الاصطفاك	الطول الضمني	نسق للكتابة	نوع الثابتة	كود
بايتة		EBCDIC	سنة	C
بايتة		ثنائي فاصلة ثابتة	سلس عشري	X
بايتة		ثنائي	ثنائي	B
كلمة	كلمة واحدة	كلمة ثنائية فاصلة ثابتة	عشري	F
نصف كلمة	نصف كلمة	نصف كلمة فاصلة ثابتة	عشري	H
كلمة	كلمة واحدة	فاصلة متحركة ودقة بسيطة	عشري	E
: كلمة مزدوجة	كلمتان	فاصلة متحركة ودقة مضاعفة	عشري	D
: كلمة مزدوجة	4 كلمات	فاصلة متحركة ودقة رباعية	عشري	L
بايتة		عشري موزع	عشري	Z
بايتة		عشري مكثف	عشري	P

جدول 1.8

في المكنة تُحصر الثوابت في حدود البايته ، نصف الكلمة ، الكلمة أو الكلمة المزدوجة حسب نوعها ما عدا في الحالة التي نُحلّد فيها طولها ( أو نستعمل مُعدّلًا للطول ) .  
 $m$  هو مُعدّل طول الثابتة ، ويمكن أن يكون :

- أ - مُعدّل طول ضمني يكتب على شكل  $Ln$  حيث  $n$  هو عدد البايتات في التمثيل الداخلي . إن وجود مُعدّل للطول يُصَفّر قاعدة الاصطفاك الضمنية .
- ب - مُعدّل للحصر يكتب على الشكل التالي :  $Sn$  .

مُعدّل الحصر يقوم بإجراء إزاحة لـ  $n$  بته إلى اليسار إذا كانت  $n$  إيجابية ، وإلى اليمين إذا كانت  $n$  سلبية . أي يقوم بإجراء ضرب أو قسمة صحيحة على  $2^n$  .  
 مُعدّل الحصر ، ويدهى أيضا المقياس ، يُطبّق على الثوابت  $E$  ،  $D$  و  $L$  .

' $n$ ' هي الثابتة المحدّدة بين فاصلتين 'علين' ( ' ) . الثوابت يمكن أن تكون محدّدة بإشارة ، فاصلة عشرية وبأس ( قوة ) يُرمز إليه بالحرف  $E$  . الأمثلة التالية تُظهر لنا مختلف الإمكانيات . وهناك جدول في الملحق يُوجّز لنا سميزات الثوابت .

LOC	SUBJECT CODE	ADDR1. ADDR2	STMT	SOURCE STATEMENT	CONST	CSECT	PRINT DATA
000230			1				
			2				
			3				
			4				
			5				
			6				
			7				
			8				
			9				
			10				
			11				
			12				
			13				
			14				
			15				
			16				
			17				
			18				
			19				
			20				
			21				
			22				
			23				
			24				
			25				
			26				
			27				
			28				
			29				
			30				
			31				
			32				
			33				
			34				
			35				
			36				
			37				
			38				
			39				
			40				
			41				
			42				
			43				
			44				
			45				
			46				
			47				
			48				
			49				
			50				
			51				
			52				
			53				
			54				
			55				
			56				
			57				
			58				
			59				
			60				
			61				
			62				
			63				
			64				
			65				
			66				
			67				
			68				
			69				
			70				
			71				
			72				
			73				
			74				
			75				
			76				
			77				
			78				
			79				
			80				
			81				
			82				
			83				
			84				
			85				
			86				
			87				
			88				
			89				
			90				
			91				
			92				
			93				
			94				
			95				
			96				
			97				
			98				
			99				
			100				



## 2.8 . ثوابت العنوان<sup>(1)</sup>

إن تعريف ثابتة - عنوان يعني حجز مكان من الذاكرة لتخزين عنوان أحد العناصر . نشير هنا إلى بعض المفاهيم الأساسية . العنوان الفعلي « أي العنوان الحقيقي لأحد العناصر هو غير معروف إلا عند شحن البرنامج في الذاكرة . لذا فمن غير الممكن ، في مرحلة التأويل والتجميع ، أن يكون بتصرفنا العنوان الفعلي الخاص بالرمز . نبلغ الرمز بواسطة الإزاحة نسبة إلى مضمون مرصف القاعدة .

في بعض الأحيان يبدو من غير الممكن بلوغ أحد الرموز التي لا تنتمي إلى الزجلة التي تكون في طور المعالجة من قبل المؤول . هذه هي الحالة ، مثلاً ، عندما نرغب بإجراء تفريع إلى برنامج - ثانوي مؤول ومترجم على حدة . الحل يقوم إذاً ، بالنسبة للمؤول ، على بلوغ مباشر بسبب وجود كلمة ، تدعى ثابتة - عنوان ، يقوم الشاحن (Loader) بملئها بشكل مناسب .

مثلاً :

نرغب « للتفريع إلى المرصف 15 ، شحنه بعنوان نقطة الدخول P1 لبرنامج - ثانوي مؤول على حدة . سنحفظ ، في الزجلة المُنَادِيَة ، كلمة تدعى هنا ADRP1 سيتم تعريفها ككتابة عنوان خارجية . والمؤول سيقوم بإعدادها وتصغيرها ، كما سيقوم الشاحن بتخزين العنوان الفعلي P1 في داخلها . العنوان P1 سنحصل عليه إذاً في المرصف 15 بواسطة التعليمة :

. L 15, ADRP1

إن نسق تعريف ثابتة العنوان هو التالي :

عامل	كود - العملية	رمز
d t m l(c)	DC	[ وسم ]

نسق هذا الأمر لا يتعمز عن نسق تعريف الثوابت إلا بتبديل الفواصل العليا بالأهلة .

- d هو عامل الإزدواجية ، وإذا جرى إسماله فإنه يعادل 1 .

- t هو كود نوع الثابتة .

(1) دراسة هذه الفقرة للفترة للفهم الكامل يمكن أن يقر عنها عند القراءة الأولى .

وقد يكون A ، Y ، S ، V أو Q ( النوع Q ليس متوفرأ سوى تحت النظام OS ) . النوعان A و Y يسمحان بتعريف الثوابت بواسطة تعابير بسيطة أو مركبة ، مطلقة أو متقولة . القيمة ثابتة العنوان عددة لجهة اليمين في كلمة ( نوع A ) أو نصف كلمة ( نوع Y ) . الثوابت من نوع S تسمح بتجزئين عنوانين بشكل قاعلة وإزاحة على نصف كلمة . ولا يمكنها أن تعرف في نص حر في . تستعمل الثوابت من نوع V لتعريف عنوانين خارجية من نوع : إسم برنامج ثانوي .

- m هو عبارة عن معدل الطول الضمني . وجود المعدل يؤدي إلى إلغاء قاعلة الاصطاف الأوتوماتيكية (alignment) .
- C هو عبارة عن الثابتة نفسها مكتوبة بدخل أهلة . الأمثلة في الصفحة 75 تعرض وتعرف كل نوع من الثوابت .

استعمال ثابتة العنوان :

تستعمل :

- لشحن عنوان في مصرف .

- لاجراء وصلات بين البرنامج والبرنامج الثانوي .

ويستم درس ذلك في الفصلين 20 و 21 .

### 3.8 . أمر حجز مواقع من الذاكرة

هذا الأمر هو عبارة عن توجيه يسمح بحجز موقع من الذاكرة دون إعداد أو تهيئة مضمونه عند التأويل . هذا الأمر يؤدي إذا إلى زيادة مضمون عداد المواقع . ويسمح بتسمية للمناطق المحددة ويبلوغها رمزياً : النحو ، القريب من نحو التوجيه DC ، هو التالي :

عمل	كود العملية	رمز
d t m	DS	[ وسم ]

- d مُعامل الإزدواجية ، وهو اختياري . وإذا كان صفراً فهو يسمح بزيادة عداد المواقع حتى حدود نصف كلمة ، كلمة أو كلمة مزدوجة حسب نوع : المرتبطة بالمنطقة . هذه الخصوصية تستعمل كثيراً ونوضحها في الأمثلة والأمثلة . سنشير هنا ، إلى أنه مع وجود عامل لإزدواجية يعادل صفراً ، فإن الوسم الموجود في منطقة الرمز هو مخزن في جدول الرموز .



٤ - يُحدّد نوع المنطقة أي بالتحديد كما جرى بالنسبة للأمر DC . وهو إلزامي ويُحدّد التطير الضمني .

m هو معدّل الطول ويكتب Ln ، حيث n هو طول المنطقة بالبايتات . كما بالنسبة للأمر DC فهو اختياري . وجوده يلغي فعل الإصطاف الضمني . سنشير هنا إلى أن الطول الأقصى للثابتة من نوع سلسلة السهات المُحدّدة في الأمر DC هو 256 بايتة ، وإستعمال النظام OS يسمح بـ 65535 بايتة .

لتسهيل صيانة البرامج سنستعمل : ETIQ DSOH لتعريف نقاط التفرع .  
قدر المستطاع سنفضل إستعمال الأمر DC عن الأمر DS الذي يقوم بإعداد المنطقة بقيمة عابدة ستكون مرئية في عملية DUMP (دلق) .

#### 4.8 . توجيه التعادل EQU

يسمح بتعريف رمز وإعطائه قيمة مطلقة أو عوّلَة ويكتب على الشكل التالي :

تعبير مطلق أو عوّل	EQU	رمز (Symbol)
--------------------	-----	--------------

سنشير هنا إلى أن وجود الرمز هو إلزامي . لا يحجز التوجيه أي موقع من الذاكرة ولا يقوم سوى بإنشاء رمز جديد في جدول الرموز . ويمكن أن يكون موجوداً في أي موقع من البرنامج ويُستخدم :

1- لاستعمال أسماء بدلاً من القيم . تجري العادة مثلاً على كتابة :

R0	EQU	0
R1	EQU	1
...	...	...
R15	EQU	15

كما يسمح ، منذ البداية ، ببلوغ المرافص بواسطة الأسماء R0 ، R1 ، ... ، R15 . بدلاً من القيم 0 ، 1 ، ... ، 15 . هذا ما يؤدي إلى فائدة ووضوح في العمل ولكن أيضاً إلى إمكانية إيجاد مراجع المرافص بسهولة لأنها ستظهر في جدول الرموز وفي البلوغ التصالي .

2- لتخصيص قيمة جديدة علّقة داخل البرنامج لرمز معين ، أي معرّف خلال الأمطر السابقة .

RD	EQU	0	
REGD	EQU	RD	( رمز مطلق )
---	---	---	
DEB	LR	R1,R2	
---	---	---	
DEBUT	EQU	DEB	( رمز محوّل )
---	---	---	
ZONE	DS	4F	
Z1	EQU	ZONE+12	( تمييز محوّل )

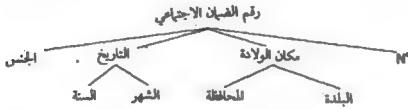
3- لحساب التعابير حيث القيمة مجهولة في لحظة الكتابة أو صيغة الحساب وتخصيص رمز لها .

EXPRES	EQU	A-(B+C)/5-D	
ETIQ	EQU	*	( قيمة عدد للواقع )

### تمارين

تمرين 1.8 - ولّد ، بواسطة تعريف ثابتة مخصصة ، منطقة من الذاكرة بحجم 100 بايتة تحتوي على سلسلة من 100 عدد صحيح طبيعي . نفس السؤال لمنطقة بحجم 100 كلمة .

تمرين 2.8 - عرّف حمز من الذاكرة لاستيعاب رقم الضمان الاجتماعي ( 13 صمة ) مع وصف للهيكلة التالية .



وذلك بفحص الخاصية - طول لكل معرف مذكور .

تمرين 3.8 - باستعمال الأمر ORG ( فقرة 3.20 ) ، مطلوب تعريف منطقة من الذاكرة يمكن أن تستوعب إما ثمانية ( 8 أرقام عشرية موسّعة ) أو كمية ( 4 أرقام عشرية موسّعة ) ، أورياً (عرجاً صحيحاً بفاصلة ثابتة) ونصاً من 10 سيات . يتعلّق ذلك بإعادة تعريف من نوع REDEFINES بلغة كوبول .

## 9 كتابة العناوين بلغة المؤول

### 1.9 . قاعدة ضمنية ، قاعدة جلية

في جسم التعليقات الآلية ، فإن العناوين المحولة تكون ممثلة بواسطة مرصفي قاعدي ، وإزاحة ومرصفي دليل ( حالة النسق RX ) . عند كتابة التعليقات - الآلية بلغة المؤول ستقوم بإيجاد ثلاثة متأثرات . لقد لاحظنا حتى الآن أنه كان يوجد ستة أنسقة مختلفة للتعليقات الآلية . إضافة لذلك ، وفي لغة المؤول ، فإن كتابة منطقة العوامل ( منطقة العناوين والنوابت ) ستتغير حسب نسق المكتبة .

لنأخذ تعليمة شحن المرصيف 3 (LOAD) من خلال مضمون عنوان معين . لنفترض إن المرصيف 15 قد جرى إختياره كمرصيف قاعدي ، وإن العنوان موضع السؤال هو موجود على مسافة 512 ( في القاعدة العاشرة ) من العنوان القاعدي وهو مؤشر بواسطة المرصيف 5 . التعليمة - الآلية سيكون لها الشكل التالي :

5	B	3	5	F	2	0	0
CDP	R <sub>1</sub>	R <sub>2</sub>	B <sub>2</sub>		D <sub>2</sub>		

سيكون بإمكاننا البرمجة بلغة المؤول أن يكتب التعليمة على الشكل التالي :  
(5, 15), 512, L 3 . القاعدة 15 هي هنا مسجلة بشكل واضح . لا نرى بهذا الشكل الفائدة الرمزية من لغة المؤول .

لتأمين بساطة أكبر فإن المؤول يسمح بعدم ذكر القاعدة في منطقة العوامل التابعة للتعليمة . يكفي لذلك أن نصريح ، بواسطة التوجيه 15, USING \* ، أن التعليقات التالية يجب أن تؤول ( تجمّع ) مع المرصيف 15 كقاعدة . الفائدة الأولى هي السهولة بتعديل مرصيف القاعدة دون إعادة كتابة جميع التعليقات . كذلك ، فإن الإزاحة ومرصيف المؤشر يمكن أن يتم تمثيلها بشكل رمزي عند الحاجة . هكذا ، فلنأخذ العنوان المحوّل ALPHA الموجود على المسافة 512 بايتة من العنوان القاعدي . ولنشحن في

المرصف 3 مضمون العنوان ALPHA للمؤشر بواسطة المرصف 5 . بإمكاننا كتابة التعليقات التالية بلغة المؤول :

- بتحليل القاعدة بشكل واضح : (5,15) L 3,512 .
- أو (5) ALPHA , L 3 ، القاعدة هي ضمناً مرتبطة بـ ALPHA وعُدّة بواسعته للمؤول حسب التوجيه USING . يوجد عدة إمكانيات لكتابة منطقة العوامل ، وهذا ما ستقوم بشرحه الآن .

## 2.9 . كتابة العوامل

في الإعتبرات التالية D ، X ، B ، R ، M و L تمثل على التوالي الإزاحة ، رقم مرصف المؤشر ، رقم مرصف القاعدة ، رقم المرصف العام ، قناع ( موجود في العملية ) والطول . الدلائل 1 ، 2 و 3 هي مرتبطة بمختلف المتغيرات . جميع هذه الرموز يجب أن تكون عبارة عن تعابير مطلقة . S ستمثل تعبيراً متحولاً يمكن أن يتحول عملياً إلى رمز واحد . وتحدد أكثر للمرصف القاعدي ، فإن عوامل ( متغيرات ) التعليقات يمكن أن تُكتب بلغة المؤول ، حسب النسق ، على الشكل التالي :

النسق	المعاملات
RR	$R_1, R_2$
RX	$R_1, D_2(X_2, B_2)$
RS	$\left\{ \begin{array}{l} R_1, R_2, D_2(B_2) \\ R_1, M_2, D_2(B_2) \end{array} \right.$
SI	$D_1(B_1), I_2$
SS	$\left\{ \begin{array}{l} D_1(L, B_1), D_2(B_2) \\ D_1(L_1, B_1), D_2(L_2, B_2) \end{array} \right.$
S	$D_2(B_2)$

## جدول 1.9

العنوان المحوّل هو دائماً العنوان المحسوب في لحظة تنفيذ الجمع  $D+X+B$  للتعليقات ذات النسق RX أو  $D+B$  للتعليقات RS ، SI أو SS . العوامل  $D(X, B)$  ،  $D(B)$  ، و  $D(L, B)$  يمكن أن تُستبدل بواسطة العوامل حيث رقم المرصف القاعدي والإزاحة سيتم حسابها بواسطة المؤول . وستكتب إذاً على الشكل التالي : S ،  $S(L)$  ، أو  $S(X)$  .

الجدول التالي يعرض لنا مختلف إمكانيات كتابة هذه المعاملات حسب نسق

التعليمة . منشير هنا إلى أنه بداخل الأهلة ، وفي الشككين مع قاعدة ضمنية أو جلية ، لا يمكن أن نجد سوى التعابير المطلقة حيث المعنى الأساسي ، الدليل أو الطول يتعلق بنسق التعليمة وبالطبيعة مطلق أو عَوَّل للتعبير المذكور على يسار الأهلة .

أمثلة :

ABS و TRANS هما تعبيران مطلقان وعَوَّلان . ABS1 (ABS2) في التعليمة RS يمكن أن تُفهم وكأنها D(B) . ABS1 (TRANS) هي مغلوبة مهما يكن النسق ، TRANS (ABS1) يجب أن تُفهم كأنها S(X) في التعليمة RX وكأنها S(L) في التعليمة . SS

نسق التعليمة	الكتابة بتعابير مطلقة قاعدة جلية	الكتابة بتعابير عَوَّل قاعدة ضمنية
RS et SI	D(B)	S
SS	D(L,B) D(L,B) (1) D(B)	S(L)
RX	D(X,B)	S(X) S

## جدول 2.9

حالات خاصة

X أو B يعادل صفرًا .

D(0) يمكن أن يكتب D

D (0,B) يمكن أن يكتب D(.B)

D (X,0) يمكن أن يكتب D(X.) أو D(X) . ( أمثلة أنظر صفحة 82 ) .

## 3.9 . قواعد الاصطفااف أو التراصف

مع أن أوالية العنونة تسمح بعنونة البايئة ، فإن عناوين متأثرات التعليمة يجب أن تخضع لبعض قواعد التوافق . قواعد كهذه هي موجودة على جميع المكتات .

تستعمل التعليقات متأثرين قد يكونان عبارة عن مرصف وعنوان من الذاكرة أو عناوين من الذاكرة . نحلّد القواعد حسب المعطيات التي تُعالجها التعليقات .

بالنسبة للتعليقات التي تُعالج كليات - مزدوجة ، كليات أو نصف - كليات ، فإن

(1) الطول هو ضمني ، المَوَّل ينظر الخاصة - طول . الطول المَوَّل هو دوماً الطول الفعلي ناقص واحد .

LOC	OBJECT CODE	ADDR1	ADDR2	STAT	SOURCE STATEMENT
00000	0000			1	PRINT DATA
00000	0000			2	START 0
00000	0000			3	* SEQUENCE D'ENTRÉE
00000	0000			4	319. 1.1.12(13)
00000	0000			5	USING 0.12
00000	0000			6	
00000	0000			7	4.3.3.3.3.3
00000	0000			8	BC 10.1.1.1.1.1
00000	0000			9	BC 10.1.1.1.1.1
00000	0000			10	BC 10.1.1.1.1.1
00000	0000			11	BC 10.1.1.1.1.1
00000	0000			12	BC 10.1.1.1.1.1
00000	0000			13	BC 10.1.1.1.1.1
00000	0000			14	BC 10.1.1.1.1.1
00000	0000			15	BC 10.1.1.1.1.1
00000	0000			16	BC 10.1.1.1.1.1
00000	0000			17	BC 10.1.1.1.1.1
00000	0000			18	BC 10.1.1.1.1.1
00000	0000			19	BC 10.1.1.1.1.1
00000	0000			20	BC 10.1.1.1.1.1
00000	0000			21	BC 10.1.1.1.1.1
00000	0000			22	BC 10.1.1.1.1.1
00000	0000			23	BC 10.1.1.1.1.1
00000	0000			24	BC 10.1.1.1.1.1
00000	0000			25	BC 10.1.1.1.1.1
00000	0000			26	BC 10.1.1.1.1.1
00000	0000			27	BC 10.1.1.1.1.1
00000	0000			28	BC 10.1.1.1.1.1
00000	0000			29	BC 10.1.1.1.1.1
00000	0000			30	BC 10.1.1.1.1.1
00000	0000			31	BC 10.1.1.1.1.1
00000	0000			32	BC 10.1.1.1.1.1
00000	0000			33	BC 10.1.1.1.1.1
00000	0000			34	BC 10.1.1.1.1.1
00000	0000			35	BC 10.1.1.1.1.1
00000	0000			36	BC 10.1.1.1.1.1
00000	0000			37	BC 10.1.1.1.1.1
00000	0000			38	BC 10.1.1.1.1.1
00000	0000			39	BC 10.1.1.1.1.1
00000	0000			40	BC 10.1.1.1.1.1
00000	0000			41	BC 10.1.1.1.1.1
00000	0000			42	BC 10.1.1.1.1.1
00000	0000			43	BC 10.1.1.1.1.1
00000	0000			44	BC 10.1.1.1.1.1
00000	0000			45	BC 10.1.1.1.1.1
00000	0000			46	BC 10.1.1.1.1.1
00000	0000			47	BC 10.1.1.1.1.1
00000	0000			48	BC 10.1.1.1.1.1
00000	0000			49	BC 10.1.1.1.1.1
00000	0000			50	BC 10.1.1.1.1.1
00000	0000			51	BC 10.1.1.1.1.1
00000	0000			52	BC 10.1.1.1.1.1
00000	0000			53	BC 10.1.1.1.1.1
00000	0000			54	BC 10.1.1.1.1.1
00000	0000			55	BC 10.1.1.1.1.1
00000	0000			56	BC 10.1.1.1.1.1
00000	0000			57	BC 10.1.1.1.1.1
00000	0000			58	BC 10.1.1.1.1.1
00000	0000			59	BC 10.1.1.1.1.1
00000	0000			60	BC 10.1.1.1.1.1
00000	0000			61	BC 10.1.1.1.1.1
00000	0000			62	BC 10.1.1.1.1.1
00000	0000			63	BC 10.1.1.1.1.1
00000	0000			64	BC 10.1.1.1.1.1
00000	0000			65	BC 10.1.1.1.1.1
00000	0000			66	BC 10.1.1.1.1.1
00000	0000			67	BC 10.1.1.1.1.1
00000	0000			68	BC 10.1.1.1.1.1
00000	0000			69	BC 10.1.1.1.1.1
00000	0000			70	BC 10.1.1.1.1.1
00000	0000			71	BC 10.1.1.1.1.1
00000	0000			72	BC 10.1.1.1.1.1
00000	0000			73	BC 10.1.1.1.1.1
00000	0000			74	BC 10.1.1.1.1.1
00000	0000			75	BC 10.1.1.1.1.1
00000	0000			76	BC 10.1.1.1.1.1
00000	0000			77	BC 10.1.1.1.1.1
00000	0000			78	BC 10.1.1.1.1.1
00000	0000			79	BC 10.1.1.1.1.1
00000	0000			80	BC 10.1.1.1.1.1
00000	0000			81	BC 10.1.1.1.1.1
00000	0000			82	BC 10.1.1.1.1.1
00000	0000			83	BC 10.1.1.1.1.1
00000	0000			84	BC 10.1.1.1.1.1
00000	0000			85	BC 10.1.1.1.1.1
00000	0000			86	BC 10.1.1.1.1.1
00000	0000			87	BC 10.1.1.1.1.1
00000	0000			88	BC 10.1.1.1.1.1
00000	0000			89	BC 10.1.1.1.1.1
00000	0000			90	BC 10.1.1.1.1.1
00000	0000			91	BC 10.1.1.1.1.1
00000	0000			92	BC 10.1.1.1.1.1
00000	0000			93	BC 10.1.1.1.1.1
00000	0000			94	BC 10.1.1.1.1.1
00000	0000			95	BC 10.1.1.1.1.1
00000	0000			96	BC 10.1.1.1.1.1
00000	0000			97	BC 10.1.1.1.1.1
00000	0000			98	BC 10.1.1.1.1.1
00000	0000			99	BC 10.1.1.1.1.1
00000	0000			100	BC 10.1.1.1.1.1

عناوين المتأثرات يجب أن تُصَفَّ حسب الحدود المناسبة . أما تلك التي تعالج السمات فلا يوجد أية مشكلة بالنسبة لها . إنَّ عدم المحافظة على هذه القواعد يؤدي إلى حدوث مشكلة في المؤول ( أنظر المثل السطر 33 ) ، فهو يؤدي عند التنفيذ إلى انقطاع من نوع «Specification» ( تميز ) . التعليقات يجب أيضاً أن تُصَفَّ في حدود نصف كلمات .

## تمارين

- تمرين 1.9 - للتعليقات أدناه :
- 1- إفحص إذا كانت العناصر التي تُولف المتأثرات هي مُطلقة أو محوَّلة .
- 2- باعتبار النسق المرتبطة بكل تعليمة نستخلص ، فقط حسب المعايير النحوية ، إذا كانت التعليقات صحيحة .
- 3- قم بإجراء تأويل التعليقات الصحيحة .

	CSECT		
	USING	*,12	مرصف القاعدة = 12
ADDBASE	L	B,D	RX
	L	3,D(3)	RX
	LR	A,D	RR
	ST	D,X'4'(3,C)	RX
	L	A,B'1011'(3)	RX
	L	D,E(B)	RX
	L	A,E(B)	RX
	MVC	A(B,C),D	SS
	MVC	E(L'D),D	SS
	L	2,D+L'D	RX
A	EQU	0	
B	EQU	1	
C	EQU	10	
D	DS	5F	
E	DS	12F	
	END		

## 10 . التعليقات بلغة المذول

### معمولات

سنقوم بدراسة التعليقات - الآلية حسب نوع التمثيل الداخلي الذي تُعالجه هذه التعليقات . من البديهي أن تكون التعليقات الحسابية العشرية ، مثلاً ، بدون معنى إلا عندما نُقدِّم لها معطيات مكوَّنة عشرياً . مثلاً ، من الواضح أن المرافف المبلوغة بالتعليقات المتحركة هي مرافف متحركة .

سنبداً بالتعليقات التي تعمل على المرافف العامة ، ولكن في البداية يجب عرض الترميز المعتمد .

#### 1.10 . الترميز

سيتم تحديد التعليقات - الآلية حسب النسق التالي :

الرمز	كود العمليات	النسق	العوامل	كود - العملية الرمزي
LOAD (S <sub>2</sub> ) → R <sub>1</sub>	COP=68	RX	R <sub>1</sub> , D <sub>2</sub> (X <sub>2</sub> , B <sub>2</sub> )	L

تشير العوامل إلى العناوين مع قاعدة محدَّدة بشكل جلي . أمَّا الشروحات فتذكر هذا العنوان بشكل رمزي . فإذا S<sub>2</sub> متبعي العنوان المحسوب بإضافة مضمون المرافف القاعدية والمؤشر إلى الإزاحة ، في المجموع فإن  $S_2 = D_2 + B_2 + X_2$  بالنسبة للتعليقات RX و  $S_2 = D_2 + B_2$  للباقية .

سنجد في الحيز مُعاملات أو في الشروحات الرموز التالية :

R<sub>1</sub>, R<sub>2</sub> هي عبارة عن أرقام المرافف التي يمكن أن تُستبدل بالتعابير المطلقة .

D قيمة الإزاحة بالنسبة إلى العنوان القاعدي .

X رقم المرافف المؤشر المُستعمل .

B رقم المرافف القاعدي .

M قناع من أربع بتات موجود في التعليمة .

I قيمة فورية موجودة في التعليمية .  
CO عدّاد البرنامج (Program counter) .  
S عنوان رمزي ، تعبير قابل للتحويل :

$$S = D_1 + X_2 + B_2$$

$$S = D_1 + B_2 \text{ أو}$$

(S) مضمون العنوان S .

→ رمز للتخصيص ، أي نسخ منطقة في أخرى دون تهديم المنطقة الأصلية . مثلاً :  
(S) → R<sub>1</sub> يعني نسخ مضمون المرصف R<sub>1</sub> في المنطقة من الذاكرة بالعنوان S . لن  
نستعمل أبداً الترميز (R) للإشارة إلى مضمون المرصف R لأنه لن يوجد أي  
إهام ، في حالة المرصف يتعلّق ذلك دائماً بالمضمون بينما يجب التمييز بين الاسم  
S للذاكرة ومضمونها .

((S)) من الممكن استعمال هذا التعبير للإشارة إلى أن مضمون العنوان S هو نفسه المعتمد  
كعنوان نأخذ منه المضمون .

CC يعني كود - الشرط .

الدلائل (indices) الدلائل 1 ، 2 ، 3 تُرجع إلى الحقول المرتبطة بالتعليمية الآلية ( فقرة  
1.5 ) .

R<sub>1</sub> (24-31) تعني البتات 24 إلى 31 من المرصف رقم R<sub>1</sub> .

R<sub>1</sub> ، R<sub>1+1</sub> تعني المرصف المزدوج المؤلف من المراصف ذات الرقم R<sub>1</sub> و R<sub>1+1</sub> . R<sub>1</sub> يكون  
رقماً مزدوجاً .

العناوين (addresses) تشير إلى أن العناوين تعني البايته من اليسار لمنطقة ما ، وإن البتات  
من الكلمة ، من مرصف ... هي مرقّمة من اليسار إلى اليمين انطلاقاً  
من 0 .

(370) تشير إلى أن التعليمية غير موجودة إلا على الكلمة 370 :

2.10 . كود العمليات الحرفية التذكيرية

كتابة كود - العمليات الرمزية يخضع إلى قواعد من المفيد الإشارة لها هنا . إن كود  
العملية يترجم الفعل المطلوب إجراؤه . السمة الأولى ( أحياناً السمتان الأوليان ) هي  
بداية الفعل الذي يُعبّر عن العمل .

مثلاً :

A	Add	جمع
L	LOAD	شحن

ST	STore	خزن
MVC	MoVe	نقل

الأحرف التالية هي معدلات (1) أو أنها تُعَيِّن نوع المعطيات المُعالِجة (2) أو أيضاً النسق RR أو SI للتعليقات (3) .

أمثلة :

(1)	AL	جمع منطقي Add Logical
(2)	CVB	تحويل إلى ثنائي ConVert Character
(2)	AE	جمع معطيات من نوع بفاصلة متحركة قصير Add données de type E (flottant court)
(2)	MVC	نقل السيات MoVe Characters
(2)	AD	جمع معطيات من نوع D
(3)	LR	شحن بنسق RR
(3)	LPR	شحن إجماعي بنسق RR
(3)	MVI	شحن مباشر بنسق SI

## 11 . الحلب بغضلة شابعة والحركات

- 1.11 . تعليقات الشحن والتخزين في المرافف العامة  
هذه هي التعليقات التي تنسخ المتأثر في أحد المرافف :  
« عنوان المتأثر . رقم المرفف LOAD »  
وتنسخ مضمون المرفف في الذاكرة على عنوان معين :  
« عنوان . رقم المرفف STORE »

هذه العمليات لا تؤثر على المتأثر الأساسي . بعض التعليقات تؤدي إلى تركيز كود الشرط CC ، لموقعين ثنائيين يتميان إلى PSW ( فصل 4 ) ، تبعاً لإشارة المتأثر المنقول حسب الإتفاق التالي :

- بعد العملية فإن CC متركز على <sup>(1)</sup> :  
- 0 إذا كانت النتيجة صفراً .  
- 1 إذا كانت النتيجة سلبية .  
- 2 إذا كانت النتيجة إيجابية .  
- 3 إذا كان هناك زيادة عن السعة (overflow) .

الزيادة عن السعة تؤدي عادة إلى إنقطاع في تنفيذ البرنامج . أي أنه سيحدث خطأ يُعالجه نظام التشغيل . يوجد برنامج ، يدعى برنامج إنقطاع «fixed point overflow» ، يعطي العلاج للمستعمل ويوقف العمل في تنفيذ البرنامج بنهاية غير طبيعية . بإمكان المبرمج أن يقوم بتجنب عملية الإنقطاع هذه في بعض الحالات بتركيز البتات المناسبة لقناع البرنامج في PSW .  
وستدرس هذا الأمر لاحقاً ( الفصل 19 ) .

---

(1) هذا الاتفاق هو صالح فقط للتعليتين LOAD وSTORE وبعض التعليمات الأخرى . وسنرى كيف يتم تركيز CC لكل مجموعة تعليمات .

المتأثر 1 هو ذاتياً مرصِف ، والمتأثر الثاني يُمكن أن يكون مرصِفاً ، نصف كلمة أو كلمة - ذاكرة .

من المهم أن نشير إلى أن التأثيرات الموجودة على العناوين المشار إليها بواسطة S يجب أن تُحصر في حدود كلمات أو نصف - كلمات حسب التعليمات .

LR	$R_1, R_2$	RR	COP=18	LOAD
				$R_2 + R_1$
L	$R_1, D_2(X_2, B_2)$	RX	COP=5B	LOAD
				$(S_2) + R_1$
		CC	لا يتغير	

LH	$R_1, D_2(X_2, B_2)$	RX	COP=4B	LOAD HALFWORD
				$(S_2) + R_1$

يُعتبر المتأثر الثاني كمتد صحيح بإشارة ويطول 16 بت . يُوسَّع إلى 32 بت قبل التحويل. CC لا يتأثر .

LCR	$R_1, R_2$	RR	COP=13	LOAD COMPLEMENT
				$R_2 + R_1$

يُجْزَأ عكس (مكْمَل إلى 2)  $R_2$  في  $R_1$  . overflow إذا أكملنا العدد السليبي الأقصى . يوضع CC حسب الإشارة النهائية لـ  $R_1$  .

LPR	$R_1, R_2$	RR	COP=10	LOAD POSITIVE
				$R_2 \rightarrow R_1$

القيمة المطلقة لـ  $R_2$  → القيمة المطلقة لـ  $R_1$  إذا أكملنا العدد السليبي الأقصى . يركّز CC على 0 ، 2 أو 3 حسب النتيجة .

LNR	$R_1, R_2$	RR	COP=11	LOAD NEGATIVE
-----	------------	----	--------	---------------

المكْمَل إلى 2 للقيمة المطلقة لـ  $R_2$  يُجْزَأ في  $R_1$  . لن يحدث overflow . CC يركّز على 0 أو 1 .

LTR	$R_1, R_2$	RR	COP=12	LOAD AND TEST
				$R_2 + R_1$

تعليمية شبيهة بـ LRR باستثناء كون الإشارة النهائية لـ  $R_1$  تُركّز CC .  $R_1$  يُمكن أن يكون معادلاً لـ  $R_2$  .

LM	$R_1, R_2, D_2(B_2)$	RS	COP=6B	LOAD MULTIPLE
----	----------------------	----	--------	---------------

الرواق التالي للذاكرة ، انطلاقاً من العنوان  $S_2$  مستشعن في المرافف العامة  $R_1$  ،  $R_1+1$  ،  $R_1+2$  ،  $R_1+3$  ، في هذه التعليمية يُفترض بأن يتبع المرافف 0 للمرافف 15 . هكذا :

LM 15, 1, ALPHA

مستشعن الكلمة ذات العنوان ALPHA في المرافف 15 ، وتلك ذات العنوان ALPHA+4 في المرافف 0 وهكذا دواليك . تُستخدم هذه التعليمية بشكل خاص لترميم إطار البرنامج CC لا يتأثر .

LA  $R_1, D_2(X_2, B_2)$  RX COP=41 LOAD ADDRESS

$S_2 \rightarrow R_1(8-31) \quad 0 \rightarrow R_1(0-7)$

تُخزَّن القيمة ذات العنوان  $S_2$  في البتات من 8 إلى 31 من المرصف  $R_1$ .  
يتم تصفير البتات من 0 إلى 7 . وتطبق هنا قواعد حساب العنوان ،  
أي أن القيمة  $D_2 + X_2 + B_2$  تُخزَّن ( عنوان فعلي ) . من الممكن أن  
تأخذ نفس المرصف  $R_1, X_2, B_2$  أو المرصف 0 لا يؤخذ أبداً وكأنه  
قاعدة أو مرصف تأثير .  
الاستعمال : أنظر التمارين  
- شحن عنوان في مرصف ،  
- شحن عدد غير سلمي أصغر أو يعادل 4095 ( القيمة القصوى للإزاحة )  
في مرصف ،

- زيادة مضمون مرصف بقيمة أصغر أو تساوي 4095 .

IC  $R_1, D_2(X_2, B_2)$  RX COP=43 INSERT CHARACTER

$(S_2) \rightarrow R_1(24-31)$

لا يتغير  $R_1(0-23)$

يتم تخزين بابتة واحدة بعنوان  $S_2$  في  $R_1$  . CC لا يتأثر .

ICM  $R_1, M_3, D_2(B_2)$  RS COP=8F INSERT CHARACTERS UNDER MASK (370)

تُرَبط البتات الأربع من القناع  $M_3$  بالبتات الأربع للمرصف  $R_1$ .  
البايتات من  $R_1$  المرتبطة بالبتات «1» من القناع يتم شحنها مع البايتات  
المتتالية من  $S_2$  . طول التأثير الثاني يعادل عدد «1» في القناع .

يُركَّز كود الشرط :

CC = 0 : جميع البتات الداخلة هي مصفَّرة أو القناع مصفَّر ،

CC = 1 : البتة ذات الوزن الأكبر في  $S_2$  هي «1» ،

CC = 2 : البتات ذات الوزن الأكبر في  $S_2$  هي «0» ولكن جميع البتات  
الداخلة ليست صفراً .

وفي الختام فإن CC يُركَّز حسب إشارة  $S_2$  .

ST  $R_1, D_2(X_2, B_2)$  RX COP=80 STORE

$R_1 \rightarrow (S_2)$

CC والمرصف  $R_1$  يبقيان بدون تعديل .

STH  $R_1, D_2(X_2, B_2)$  RX COP=40 STORE HALFWORD

$R_1(16-31) \rightarrow (S_2)$

التأثير الثاني هو بطول 2 بابتة . CC يبقى بدون تعديل .

STM  $R_1, R_3, D_2(B_2)$  RS COP=8D STORE MULTIPLE

للمراصف العادة من  $R_1$  إلى  $R_3$  يتم تخزينها في مواقع متتالية من الذاكرة  
بدءاً من العنوان  $S_2$  . الرقم 0 للمرصف 0 مُفترض أنه يتبع الرقم 15  
بشكل يؤدي منه تنبه التعليمة ALPHA 1, 15, ST إلى تخزين

المرافق 15 ، 0 ، 1 بالمتاوين ALPHA+4 ، ... ،  
تستخدم التعليمة بشكل خاص لحفظ إطار البرنامج . CC يبقى بدون  
تغيير .

---

STC  $R_1, D_2(X_2, B_2)$  RX COP=42 STORE CHARACTER  
 $R_1(24-31) + (S_2)$   
R1 و CC يقيان بدون تعديل .

---

STCM  $R_1, M_2, D_2(B_2)$  RS COP=BE STORE CHARACTERS UNDER MASK  
(370)  
البيت الأربع من القناع M ترتبط بالأربع بالبت من للرفيف R1 .  
لما بالبت R1 ، والمختارة بوجود «1» في القناع ، فيتم تخزينها بشكل  
متراس على العنوان S2 . كود الشرط CC لا يتغير .

---

## 2.11 . التعليقات الحسابية بفاصلة ثابتة

هي التعليقات التي تعمل على معطيات ممثلة بفاصلة ثابتة . تكوّد القيم السلبية  
بواسطة المكمل إلى 2 . كما تقوم بالعمليات الأربع الأساسية بين مرصوف ومرصوف أو  
بين مرصوف وذاكرة . الضرب والجمع يستعملان مرافق مزدوجة (فقرة 1.10) . هذه  
التعليقات تؤدي إلى تعديل CC حسب إشارة النتيجة ، وحسب الإتفاق الجاري كما في  
1.11 .

CC = 0 إذا كانت النتيجة صفراً .

CC = 1 إذا كانت النتيجة سلبية .

CC = 2 إذا كانت النتيجة إيجابية .

CC = 3 إذا كان هناك overflow .

يمكن قطع التعليمة في حالة حدوث حادثة غير طبيعية ، كما يلي :

- عنوان من خارج المنطقة المخصصة .

- جهة متأثر غير صحيحة ، مرصوف مزدوج معني بشكل سيء .

- لفيض عن السعة overflow .

AR  $R_1, R_2$  RR COP=1A ADD  
 $R_1 + R_2 \rightarrow R_1$   
A  $R_1, D_2(X_2, B_2)$  RX COP=5A ADD  
 $R_1 + (S_2) \rightarrow R_1$

لا يتغير المتائر الثاني . يتم تركيز كود الشرط CC ، احتمال  
حصول overflow .

---

AH  $R_1, D_2(X_2, B_2)$  RX COP=4A ADD HALFWORD  
 $(S_2) + R_1 \rightarrow R_1$   
 التأثير (S<sub>2</sub>) هو على نصف كلمة . يُوسَّع إلى كلمة قبل العملية . يتم تركيز CC  
 احتمال حصول Overflow .

SR  $R_1, R_2$  RR COP=1B SUBTRACT  
 $R_1 - R_2 \rightarrow R_1$   
 S  $R_1, D_2(X_2, B_2)$  RX COP=6B SUBTRACT  
 $R_1 - (S_2) \rightarrow R_1$   
 التأثير التالي لا يتعدل يتم تركيز CC .

SH  $R_1, D_2(X_2, B_2)$  RX COP=4B SUBTRACT HALFWORD  
 $R_1 - (S_2) \rightarrow R_1$   
 التأثير S<sub>2</sub> هو على نصف كلمة ، يُوسَّع إلى 32 بتة قبل العملية . يتم تركيز CC .

MR  $R_1, R_2$  RR COP=1C MULTIPLY  
 $R_{1+1} \times R_2 \rightarrow R_1, R_{1+1}$   
 M  $R_1, D_2(X_2, B_2)$  RX COP=6C MULTIPLY  
 $R_{1+1} \times (S_2) \rightarrow R_1, R_{1+1}$   
 المرصف R<sub>1</sub> المذكور في التعليمة يجب أن يكون مرصفاً مزدوجاً . التأثير الأول يجب أن يكون موجوداً في R<sub>1+1</sub> ومحصوراً لجهة الشئال . النتيجة ستوضع في R<sub>1</sub> ، R<sub>1+1</sub> . لا احتمال لحصول overflow ، لا يتم تركيز CC .

MH  $R_1, D_2(X_2, B_2)$  RX COP=4C MULTIPLY HALFWORD  
 $R_{1+1} \times (S_2) \rightarrow R_1, R_{1+1}$   
 المرصف R<sub>1</sub> يجب أن يكون مرصفاً مزدوجاً ، S<sub>2</sub> يتألف من 16 بتة ويُعتبر كعدد صحيح بإشارة يُوسَّع إلى 32 بتة قبل العملية . لا يحدث overflow ولا يتم تركيز CC .

DR  $R_1, R_2$  RR COP=1D DIVIDE  
 $R_1, R_{1+1} : R_2 \rightarrow R_1, R_{1+1}$   
 الباقي  
 قيمة القسمة  
 R<sub>1</sub> هو مرصف مزدوج . يتمتع الباقي بنفس إشارة المقسوم . عندما لا تسع 32 بتة نتيجة القسمة يحدث overflow . لا يتم تركيز CC .

D  $R_1, D_2(X_2, B_2)$  RX COP=6D DIVIDE  
 $R_1, R_{1+1} : (S_2) \rightarrow R_1, R_{1+1}$   
 الباقي  
 قيمة القسمة  
 R<sub>1</sub> يجب أن يكون مرصفاً مزدوجاً . للباقي نفس إشارة المقسوم . عندما لا تسع 32 بتة نتيجة القسمة يكون هناك هفص عن السعة . لا يتم تركيز CC .

#### ملاحظات :

دراسة هذه التعليقات تسمح لنا بملاحظة إن النتيجة تُحل دائماً مكان المتأثر الأول الذي يضيع منا . بينما لا يتم تعديل المتأثر الثاني . التعليقات التي تجري على نصف كلمة تفترض توسيع نصف الكلمة إلى كلمة قبل العملية .

#### 3.11 . عمليات المقارنة بفاصلة ثابتة

تؤثر تعليمات المقارنة فقط على مضمون كود الشرط . هذه التعليقات هي خاصة حسب نوع تمثيل المعطيات المقارنة . سندرس هنا تلك المتعلقة بالفاصلة الثابتة . كما في التعليقات التي رأيناها ، فإن المتأثر الأول هو دائماً موجود في مرصف معين والمتأثر الثاني في مرصف آخر أو في الذاكرة . يجري تركيز CC حسب الطريقة التالية :

- 0 = CC إذا كان المتأثر الأول = المتأثر الثاني .
- 1 = CC إذا كان المتأثر الأول أصغر من المتأثر الثاني .
- 2 = CC إذا كان المتأثر الأول أكبر من المتأثر الثاني .
- 3 = CC لا يُستعمل .

CR	$R_1, R_2$	RR	COP=19	COMPARE
C	$R_1, D_2(X_2, B_2)$	RX	COP=59	COMPARE

المقارنة هي جبرية وتتعلق بـ 32 بتة . يتم تركيز مضمون CC .

CH	$R_1, D_2(X_2, B_2)$	RX	COP=49	COMPARE HALFWORD
----	----------------------	----	--------	------------------

يُوسّع للمتأثر الثاني إلى 32 بتة قبل المقارنة مع إنتشار بتة الإشارة . يتم تركيز CC .

#### 4.11 الجمع والطرح المنطقي

نعني بالجمع والطرح المنطقي ، عمليات تعديل مضمون CC بطريقة مختلفة عن الجمع والطرح العادي الذي رأيناه أعلاه . إضافة لذلك فإن overflow لا يؤدي إلى قطع البرنامج

يتم تركيز CC على الشكل التالي :

- 0 = CC إذا كانت النتيجة صفراً بدون مرّحل .
- 1 = CC إذا كانت النتيجة مختلفة عن 0 بدون مرّحل (no carry)
- 2 = CC إذا كانت النتيجة صفراً مع مرّحل .
- 3 = CC إذا كانت النتيجة مختلفة عن صفر مع مرّحل .

ALR $R_1, R_2$	RR COP=1E	ADD LOGICAL $R_2 + R_1 \rightarrow R_1$
AL $R_1, D_2(X_2, B_2)$	RX COP=5E	ADD LOGICAL $(S_2) + R_1 \rightarrow R_1$

---

SLR $R_1, R_2$	RR COP=1F	SUBTRACT LOGICAL $R_1 - R_2 \rightarrow R_1$
SL $R_1, D_2(X_2, B_2)$	RX COP=5F	SUBTRACT LOGICAL $R_1 - (S_2) \rightarrow R_1$

---

### 5.11 . التحريك من الذاكرة إلى الذاكرة

تتم في أغلب الأحيان بواسطة تعليمات من نوع SS . لا يوجد أي تقييد فيما يتعلق بالاصطفاف (alignment) . يمكن أن يتم تركيز الطول بشكل واضح في التعليمات : MVC ZONE 1, ZONE 2 أو ضمناً MVC ZONE 1 (L), ZONE 2 . يقوم عندها المؤول باختيار خاصية - الطول الخاصة بالمتأثر الأول L'ZONE 1 . الطول المؤول هو الطول المذكور في التعليمة ناقص 1 . يمكن للمتأثرين أن يتراكبا ، ونجد هذه الميزة مستعملة في التمرين 6.11 .

MVI $D_1(B_1), I_2$	SI COP=9Z	MOVE $I_2 \rightarrow (S_1)$
---------------------	-----------	---------------------------------

يتم تخزين البلية للباشرة  $I_2$  في  $S_1$  .

---

MVC $D_1(L, B_1), D_2(B_2)$	SS COP=DZ	MOVE
-----------------------------	-----------	------

$(S_2) \rightarrow (S_1)$  بطول L .

الحركة تتم من اليسار إلى اليمين . العملية هي غير قابلة للانقطاع عند نقل بايتين . يسمح بالترابك وفي هذه الحالة يحدد الانتباه إلى أن الحركة تجري من اليسار إلى اليمين من أجل الحصول على النتيجة .

---

MVCL $R_1, R_2$ (370)	RR COP=0E	MOVE LONG
--------------------------	-----------	-----------

نسخ المتأثر الثاني في المتأثر الأول .

$R_1 (8 - 36)$  يحتوي على عنوان المتأثر الأول ،

$R_1 + 1 (8 - 36)$  طول المتأثر الأول ،

$R_2 (8 - 36)$  عنوان المتأثر الثاني ،

$R_2 + 1 (8 - 7)$  سمة تسمية ،

$R_2 + (8 - 36)$  طول المتأثر الثاني .



الحركة تتم من اليسار إلى اليمين ، لكل بايت على حدة . العملية هي قابلة للانقطاع عند نسخ بايتين . إذا كان طول المتنبر الثاني هو أصغر من طول المتنبر الأول ، يتم تكملة المتنبر الأول بسمة تبعية . يمكن تراكب المناطق بشرط أن لا يقوم بالنسخ بتعديل بايت جرى تعديلها سابقاً .

يجري تركيز CC على الشكل التالي :

CC = 0 إذا كان كلا المتنبرين بنفس الطول ،

CC = 1 المتنبر الأول هو أقصر ،

CC = 2 المتنبر الأول هو أطول ،

CC = 3 إذا أدت عملية التطابق إلى تعديل في بايت معطلة أصلاً .

يمكن استعمال علم العملية لتفسير الماكرو .

MVN D<sub>1</sub>(L<sub>1</sub>,B<sub>1</sub>),D<sub>2</sub>(B<sub>2</sub>) SS COP=D1 MOVE NUMERIC

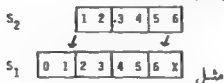
نسخ نصف - بايتات بالوزن الأضعف من (S<sub>2</sub>) في أنصاف - بايتات الوزن الأضعف من (S<sub>1</sub>) . تبقى أنصاف - البايتات بالوزن الأقوى دون تعديل . يسمح بالتراكب وهذا الصدد نمطي للملاحظة نفسها كما بالنسبة لـ MVC .

MVZ D<sub>1</sub>(L<sub>1</sub>,B<sub>1</sub>),D<sub>2</sub>(B<sub>2</sub>) SS COP=D3 MOVE ZONES

نسخ نصف بايتات بالوزن الأقوى من (S<sub>2</sub>) في نصف بايتات الوزن الأقوى من (S<sub>1</sub>) . تبقى أنصاف - البايتات بالوزن الأضعف دون تعديل . يسمح بتراكب المحزات وهذا الصدد نمطي للملاحظة نفسها كما بالنسبة لـ MVC .

MVD D<sub>1</sub>(L<sub>1</sub>,B<sub>1</sub>),D<sub>2</sub>(L<sub>2</sub>,B<sub>2</sub>) SS COP=F1 MOVE WITH OFFSET

نسخ من (S<sub>2</sub>) في (S<sub>1</sub>) مع إزاحة إلى اليسار مقدار نصف بايت . العملية تتم من اليمين إلى اليسار ، بايت بعد بايت . لا يتم تغيير آخر بايت لجهة اليمين .



## تمارين

تمرين 1.11 - ضع في الصفر الثنائي أحد المرافف ( أعط حلين لتعلية واحدة دون حجز ثوابت ) .

تمرين 2.11 - غير إشارة المرصف ( تمثيل ثنائي ) .

تمرين 3.11 - ضع جميع بتات المرصف في 1 .

تمرين 4.11 - اشحن القيمة 2048 في مرصف ، ثم القيمة 4095 ( دون حجز ثابتة ) بعد ذلك اشحن 4096 .

تمرين 5.11 - زد مضمون أحد المرافف مقدار 4 .

تمرين 6.11 - عىء منطقة بطول  $L \gg 256$  باءة بنجوم ( تعلمتان ) .

نقوم بالتفريع كل تعديل في مضمون عداد البرنامج يؤدي إلى إنقطاع في الدوران التالي للعمليات .

- هودتنا دراسة اللغات المطورة على اعتبار نوعين من الإنقطاعات في التالية :
- الإنقطاعات الإلزامية ( GOTO في لغة فورتران ) .
- الإنقطاعات للشروط ( IF ) .

في لغة المؤلف ، فإن الإنقطاعات المشروطة تنتج إما عن اختيار لقيمة مأخوذة من كود الشرط ، إما عن اختيار لقيمة مأخوذة من مرصف عام . التعليمتان BC و BCR تفحصان كود الشرط CC والتعليمات BCT ، BCTR ، BXH ، BXLE تُفحص أو تزيد من مضمون مرصف وبعد ذلك تفحص قيمته .

يمكن تنفيذ الإنقطاعات الإلزامية بواسطة BC و BCR .

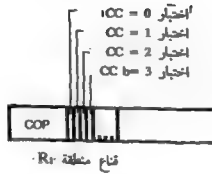
#### 1.12 . الكود - الشرط

لقد التقينا عند دراسة التعليمات السابقة . ونذكر بأنه عبارة عن مؤشر بموقعين ثنائيين ، يتمان إلى PSW (البتان، 34 ، 35 ) ويؤكزان بواسطة بطبع تعليمات حسب النتيجة الحاصلة . التعليمات الحسابية ، مثلاً ، التركيز حسب إشارة النتيجة ، تعليمات المقارنة حسب القيمة النسبية لمثليين .

الكود الشرطي CC يمكن أن يأخذ إذن أربع قيم ثنائية 00 ، 01 ، 10 ، 11 يتم مراجعتها في التعليمات بواسطة 0 ، 1 ، 2 ، 3 .

#### 2.12 . التعليمات التي تفحص الكود الشرطي (CC) : BCR و BC

هذه التعليمات تستعمل المنطقة Ri ، المكونة من أربع بتات ثنائية ، من نسقها الآلي ، ليس كرقم مرصف بل كتنوع : كل بة تعادل 1. وموجودة في هذه المنطقة تناسب اختبار إحدى القيم الأربع التي نحصل عليها بواسطة CC حسب الإتفاق التالي :



هكذا ، فالقناع المعدل 1100 (ثنائياً) سيسمح باختبار الشروط  $CC=0$  أو  $CC=1$ . الشرط المختار فعلاً يتعلّق إذاً بالتعلّمة التي أدت إلى تركيز  $CC$ .  
لقد رأينا أن  $CC$  تركّز حسب الطريقة التالية :

كود الشرط	0	1	2	3
تعلّيمات حسابية نتيجة ....	$=0$	$<0$	$>0$	ليس من السعة
تعلّيمات مقارنة متأثر أول ....	$=$	$<2^0$	$>2^0$	---

القناع المعدل لـ 1100 (أي C بالنظام السادس عشري أو 12 بالعشري ) يناسب الاختبارات التالية :

- نتيجة سلبية أو صفر بعد تعلّمة حسابية .
- متأثر أول أصغر من المتأثر الثاني بعد تعلّمة مقارنة .

BCR  $M_1, R_2$

RR COP=07 BRANCH ON CONDITION

$M_1$  هي القناع المذكور أعلاه .

بعد تنفيذ الشرط ، هناك تفرّع إلى العنوان المخزّن في  $R_2$  . وإلا سيستمر التنفيذ بالتوالي . كما يترجم حل الشكل التالي : الشرط المنفذ :  $R_2 \rightarrow CO$   
وإلا  $CO + 2 \rightarrow CO$

BC  $M_1, D_2(X_2, B_2)$

RX COP=47 BRANCH ON CONDITION

$M_1$  قناع .

إذا تم تنفيذ الشرط فيحدث تفرع إلى العنوان  $D_2 + X_2 + B_2$   
 وإلا فإن التنفيذ سيتابع بالتوالي ، مما يترجم على الشكل التالي :  
 في حال تنفيذ الشرط :  $D_2 + X_2 + B_2 \rightarrow CO$   
 وإلا  $CO + 4 \rightarrow CO$  و  $D_2 + X_2 + B_2$  عنوان التفرع ..

في لغة المؤول ، يُعَدُّ القناع  $M_1$  بواسطة تعبير مطلق ، عادة رقم عشري .  
 BC 15, ALPHA أو BCR 15,R يناسبان القناع 1111 . يتعلّق ذلك إذا  
 بالتفرع المنتظم لأنه منها تكن قيمة CC هناك تفرع .  
 BCR 0,R أو BC 0,ALPHA هي عبارة عن تعليقات دون فعل لأنه لن يتم  
 اختبار أي شرط . وهي تميّز بأنها بدون فعل .  
 الأكواد الحرفية التذكيرية الموسّعة  
 وفي النهاية كي يتم تفادي تحديد القناع الخاص ولتذكر الإضافات المذكورة أعلاه ،  
 فإن المؤول يسمح باستعمال كود حرفي حسب الشرط المفحوص .  
 ويقوم بمهمة ترجمة الكود الحرفي إلى BC أو BCR .  
 هكذا :

$B \quad D_2(X_2, B_2)$  يناسب تقريباً غير شرطي  
 BC 15,  $D_2(X_2, B_2)$

$BR \quad R_2$  يناسب تقريباً غير شرطي  
 BCR 15,  $R_2$

$BNE \quad D_2(X_2, B_2)$  يناسب تقريباً معيّناً وإلا يعادل  
 BC 7,  $D_2(X_2, B_2)$

سنجد في الملحق اللاحقة الكاملة للكود الحرفي التذكيري الموسّع . سنلاحظ إن  
 الأكواد الحرفية تتعلّق بالتعليمة التي تقوم بتركيز الكود الشرطي . من المفيد ، لوضوح  
 البرنامج ، استعمال هذه الأكواد الحرفية التذكيرية . ونركّز على كون هذه الأكواد العملية  
 لا تتناسب سوى مع 2 كود - مكنة . ونشير ، كما ذكرنا في الفقرة 2.10 ، إلى أن الأكواد  
 التي تنتهي بـ R تناسب تعليقات بنسق RR أو BCR .

3.12 . . التعليقات التي تفحص القيمة المأخوذة من مرصيف (مؤشر)  
 أربع تعليقات BCT ، BCTR ، EXH و BXLE تسمح بتعلييل مضمون

المصرف والتفرع إلى عنوان معين عندما تصبح قيمته معادلة ، أقل أو أكبر من كمية محددة .

BCTR  $R_1, R_2$  RR COP=06 BRANCH ON COUNT  
 $R_1 - 1 \rightarrow R_1$

إذا كانت  $R_1 \neq 0$  :  $CO \rightarrow R_2$  (تفرع إلى العنوان الموجود في  $R_2$ ) .  
وإلا  $CO + 2 \rightarrow CO$  (تنفيذ التعليمة التالية) .  
ملاحظة : إذا كان  $R_2$  هو المصرف 0 فالمدّ يتم بدون تفرع .

BCT  $R_1, D_2(X_2, B_2)$  RX COP=46 BRANCH ON COUNT  
 $R_1 - 1 \rightarrow F_1$

إذا :  $S_2 \rightarrow CO$  :  $R_1 \neq 0$  (تفرع إلى العنوان  $S_2$ )  
وإلا :  $CO + 4 \rightarrow CO$  (تنفيذ التعليمة التالية) .

BXH  $R_1, R_2, D_2(B_2)$  RS COP=86 BRANCH ON INDEX HIGH

1- زيادة مضمون  $R_1$  :  $R_1 + R_2 \rightarrow R_1$   
2- عندما تصبح  $R_1$  أكبر من المرجعية : تفرع . المرجعية هي  $R_{2+1}$

أ-  $R_2$  هو مصرف برقم مفرد .  
 $R_2$  هو مرجع المقارنة والزيادة .

فإنّ :  $R_1 + R_2 \rightarrow R_1$  بعد ذلك ، إذا كان  $R_1 > R_2$   
عندئذ  $CO \rightarrow S_2$  (تفرع إلى  $S_2$ )  
وإلا  $CO + 4 \rightarrow CO$  (متابعة على التوالي) .  
ب-  $R_2$  هو مصرف برقم مزدوج

نستعمل المصرف المزدوج  $R_2$  و  $R_{2+1}$   
 $R_2$  هو الزيادة و  $R_{2+1}$  هو المرجعية . إذن  $R_1 + R_2 \rightarrow R_1$  ثم  
إذا كان  $R_1 > R_{2+1}$  عندئذ  $CO \rightarrow S_2$  (التفرع إلى  $S_2$ ) .  
وإلا  $CO + 4 \rightarrow CO$  (متابعة المتتالية) .

ملاحظة :

يجب أن لا نخلط هنا بين المصطلح إشارة مع مصرف المؤشر  
للتعليقات RX .  
للمقارنة تتم جبرياً . ويتم إجمال overflow عند الجمع .

BXLE  $R_1, R_2, D_2(B_2)$  RS COP=87 BRANCH ON INDEX LOW OR EQUAL

1- زيادة  $R_1$  :  $R_1 + R_2 \rightarrow R_1$



EX R<sub>1</sub>,D<sub>2</sub>(X<sub>2</sub>,B<sub>2</sub>) RX COP=44 EXECUTE

هذه التعليمة تسمح بتنفيذ تعليمة واحدة موجودة خارج التسلسل الطبيعي للعنوان S<sub>2</sub> . بعد ذلك ، فإن العمل يُعاود بالتوالي .

يتم تنفيذ عملية « أو » متضمنة بين البتات (R1(31-15) و R2(15-8) ) تسمح بتعديل هذا الحقل من التعليمة ( رقم المرصف ، قيمة تلقائية أو طول ) . إذا كان R<sub>2</sub> هو المرصف 0 فلا يتم تنفيذ العملية « أو » (OR) . كما لا يمكن تنفيذ عملية التحويل .

تطبيق

عندما نرغب بإجراء نقل للمعلومات MVC من منطقة لا نعرف طولها إلا في لحظة التنفيذ . هذه الحالة تحدث عند معالجة التسجيلات بطول متغير ، يكون طول الفقرة موجوداً في رأسها . من الممكن إذاً تنفيذ التعليمة «MVC» . والطريقة هي التالية : شحن الطول في R1(31-15) :

BCTR R<sub>1</sub>, 0 (تنقيص 1)  
EX R<sub>1</sub>,MOVE  
..-----  
MOVE MVC -----

تُنفَّذ MVC مع الطول المطلوب دون أن يكون هناك تعديل للتعليمة في الذاكرة . التعليمة MVC لا تعمل إلا خلال مدة التنفيذ . ويمكن أن تكون موجودة في أي مكان ولكن يُفضل أن تكون EX و MVC موجودتين في نفس الصفحة من الذاكرة كي لا تقع في خطأ عتَمَل في نفس الصفحة .

تمارين :

تمرين 1.12 . أكتب متتالية التعليقات التي تسمح بتكرار N مرة إحدى عمليات المعالجة .

تمرين 2.12 . احسب مجموع عناصر جدول من الكلمات يحتوي على أعداد بفاصلة ثابتة .

تمرين 3.12 . إعكس سلسلة من البتات CH1 في CH2 .

تمرين 4.12 . نقص مضمون المرصف 1 (تعليمة واحدة) .

تمرين 5.12 . إشن مرصفاً معيناً بالعنوان الجاري زائد 2 .

## 13 . العمليات المنطقية

### 1.13 . الدوال المنطقية

يسمح الكمبيوتر IBM 360/370 بعمونة البايته ، ومن غير الممكن الإشارة إلى بته معينة داخل البايته . ولكن بسبب وجود تعليمات الإزاحة (Shift) والتعليمات المنطقية سيكون بإمكاننا إختيار أو تعديل مضمون إحدى البتات من داخل الكلمة .

العمليات المنطقية الموجودة هي «و» (AND) ، الجمع «أو» (OR) و «أو المقصورة» (EOR) . جدول العمليات المنطقية هو التالي :

A	1	0	1	0	تعليمات
B	1	1	0	0	
A AND B	1	0	0	0	NR N NI NC
A OR B	1	1	1	0	OR O OI OC
A FOR B	0	1	1	0	XR X XI XC

### 2.13 . التعليمات المنطقية

التأثيرات هي :

- مرصفتان عامتان (شكل RR) : التعليمات NR ، OR ، XR ،
  - مرصفت كلمة - ذاكرة (شكل RX) : التعليمات N ، O ، X ،
  - بايئة موجودة في التعليمة وبايئة موجودة في الذاكرة (الشكل SI عنونة مباشرة) :
  - التعليمات NI ، OI ، XI ،
  - سلسلتان من البايتات في الذاكرة (شكل SS) : التعليمات NC ، OC ، IC .
- توضع النتيجة دائماً في المتأثر a1 .
- يتم تركيز كود الشرط حسب الطريقة التالية :

CC	
0	إذا كانت النتيجة تماثل صفر
1	إذا كانت النتيجة مختلفة عن صفر

عمليات الإنقطاع الممكنة تتعلق ، كالعادة ، بمسألة العنوان : تعدُّ على المنطقة المخصصة من الذاكرة ، تعدُّ على المنطقة الممكنة من الذاكرة أو مشكلة الزيادة في مضمون المرافص المزدوجة .

التقاطع « و » (AND)

NR  $R_1, R_2$  RR COP=14 AND  
 $R_1 \llcorner \text{And} \gg R_2 \rightarrow R_1$   
 تتم العملية على 4 بايتات .

N  $R_1, D_2(X_2, B_2)$  RX COP=54 AND  
 $R_1 \llcorner \text{And} \gg (S_2) \rightarrow R_1$   
 تتم العملية على أربع بايتات .

NI  $D_1(B_1), I_2$  ■ COP=94 AND  
 $(S_1) \llcorner \text{And} \gg I_2 \rightarrow (S_1)$   
 H هي قيمة تلقائية موجودة في التعليمات . العملية تتم على بايتة واحدة .

NC  $D_1(L, B_1), D_2(B_2)$  SS COP=D4 AND  
 $(S_1) \llcorner \text{And} \gg (S_2) \rightarrow (S_1)$

العملية تتم بين منطقتين من الذاكرة بطول مشترك هو L بايتة . ويجري العملية بايتة بعد بايتة من اليسار إلى اليمين . كل شيء يسير كما لو كانت كل بايتة محسوبة وخزنة في الذاكرة قبل العبور إلى البايته التالية .

تطبيق عملي :

تصغير إحدى البتات .

الجمع « أو »

OR  $R_1, R_2$  RR COP=18 OR  
 $R_1 \llcorner \text{OR} \gg R_2 \rightarrow R_1$   
 تتم على أربع بايتات

O  $R_1, D_2(X_2, B_2)$  RX COP=58 OR  
 $(S_2) \llcorner \text{OR} \gg R_1 \rightarrow R_1$   
 تتم على أربع بايتات .

$$OI \quad D_1(B_1), I_2 \quad SI \quad COP=88 \quad OR \quad (S_1) \leftarrow OU \leftarrow I_2 \rightarrow (S_1)$$

I<sub>2</sub> هي قيمة موجودة في التعليم . تجري العملية على بايتة واحدة .

$$OC \quad D_1(L, B_1), D_2(B_2) \quad SS \quad COP=D8 \quad OR \quad (S_2) \leftarrow OR \leftarrow (S_1) \rightarrow (S_1)$$

تتم العملية على منطقتين من الذاكرة بطول مشترك هو L بايتة . ويتم بايتة بعد أخرى من اليسار إلى اليمين .

تطبيق عملي :

جعل إحدى البتات تعادل 1 .

« أو المختصرة » (EOR)

$$XR \quad R_1, R_2 \quad RR \quad COP=17 \quad EXCLUSIVE \ OR \quad R_1 \leftarrow EOR \leftarrow R_2 \rightarrow R_1$$

تتم العملية على أربع بايتات .

$$X \quad R_1, D_2(X_2, B_2) \quad RX \quad COP=57 \quad EXCLUSIVE \ OR \quad R_1 \leftarrow EOR \leftarrow (S_2) \rightarrow R_1$$

تتم العملية على أربع بايتات .

$$XI \quad D_1(B_1), I_2 \quad SI \quad COP=97 \quad EXCLUSIVE \ OR \quad (S_1) \leftarrow EOR \leftarrow I_2 \rightarrow (S_1)$$

I<sub>2</sub> هي قيمة تلقائية موجودة في التعليم . تتم العملية على بايتة واحدة .

$$XC \quad D_1(L, B_1), D_2(B_2) \quad SS \quad COP=D7 \quad EXCLUSIVE \ OR \quad (S_1) \leftarrow EOR \leftarrow (S_2) \rightarrow (S_1)$$

تجري العملية على منطقتين من الذاكرة بطول مشترك L بايتة ، وتجري بايتة بعد بايتة من اليسار إلى اليمين كما لو كانت كل بايتة قد جرى حسابها وتخزينها في الذاكرة قبل العبور إلى البايته التالية .

تطبيق عملي :

عكس البتة ، مكمل منطقي ، تصفير منطقة من الذاكرة .

### 3.13 . المقارنات المنطقية

كما في جميع العمليات المنطقية تجري معالجة جميع البتات بنفس الطريقة . لا وجود لأي تمييز للبتة ذات الوزن الأعلى . تتم المقارنة من اليسار إلى اليمين وتتوقف عند أول معادلة . يُركّز كود الشرط حسب الطريقة التالية :

( نذكر أنّ المتأثر الأول هو ذلك الذي يتم بلوغه في التعليم بواسطة المؤشر 1 .

الإنقطاعات الممكنة هي تلك المتعلقة بالمتونة وتلك المتعلقة بحدود الكلمات .

0	إذا كانت التأثيرات متساوية
1	إذا كان التأثير الأول أصغر من التأثير الثاني
2	إذا كان التأثير الأول أكبر من التأثير الثاني
3	غير مستعمل

CLR  $R_1, R_2$  RR COP=15 COMPARE LOGICAL  
مقارنة بين كامل المرافف .

CL  $R_1, D_2(X_2, B_2)$  RX COP=55 COMPARE LOGICAL  
مقارنة على أربع بايتات .

CLI  $D_1(B_1), I_2$  SI COP=95 COMPARE LOGICAL  
مقارنة منطقية مباشرة بين القيمة  $I_2$  الموجودة في التعليمة و  $(Si)$  .

CLC  $D_1(L, B_1), D_2(B_2)$  SS COP=D5 COMPARE LOGICAL  
مقارنة بين سلاسل تمتد حتى 256 بايت بطول مشترك L .

CLM  $R_1, M_2, D_2(B_2)$  RS COP=BD COMPARE LOGICAL CHARACTERS UNDER MASK  
(370)

القناع  $M_2$  ، المكوّن من أربع بتات يُختار في  $R_1$  من 0 إلى 4 بايتات تُقارن بالبايتات المتتالية إنطلاقاً من العنوان  $Se$  . البتة الأولى من القناع ، إذا كانت مملوءة بـ 1 تختار البايته الأولى من  $R_1$  وهكذا دواليك . يتم تركيز CC .

القناع المعادل لـ 1011 يختار البايتات 0 ، 2 ، 3 من  $R_1$  التي تتم مقارنتها مع ثلاث بايتات إنطلاقاً من  $Se$  . المقارنة تتم من اليسار إلى اليمين .

CLCL  $R_1, R_2$  RR COP=OF COMPARE LOGICAL LONG  
(370)

مقارنة بين سلسلتين من البايتات حيث المتولين والأطوال موجودة في المرافف المزدوجة حسب الإفق الثاني :



تجري العملية من اليسار إلى اليمين من خلال العناوين 1 و 2 . إذا لم يكن طول السلسلتين متعادلاً ، يُفترض بأن يكتمل الأقصر من اليمين بالسمة «padding» (سمة الحشو) .

العملية تتم بايتة بعد بايتة مع زيادة عناوين وتقصير الطول . وهي قابلة للانقطاع بين مقارنة بايتين . وتتوقف عند أول لا متعادلة نلتقيها أو في نهاية السلسلة مع تركيز كود - الشرط .

#### 4.13 . مقارنات منطقية خاصة

لقد قمنا هنا بتصنيف التعليقات التي ، زيادة عن وظيفتها في المقارنة ، تتمتع بعمل خاص . هذه التعليقات تركز كود الشرط بصورة مختلفة .

CS R<sub>1</sub>,R<sub>2</sub>,D<sub>3</sub>(B<sub>2</sub>) RS COP=BA COMPARE AND SWAP

(370)

مقارنة بين R<sub>1</sub> و R<sub>2</sub>

إذا : (S<sub>2</sub>) = R<sub>1</sub> محتفل (S<sub>2</sub>) → R<sub>2</sub> و CC → 0 .

إذا : (S<sub>2</sub>) ≠ R<sub>1</sub> محتفل (S<sub>2</sub>) → R<sub>2</sub> و CC → 1 .

CDS R<sub>1</sub>,R<sub>2</sub>,D<sub>3</sub>(B<sub>2</sub>) RS COP=BB COMPARE DOUBLE AND SWAP

(370)

مقارنة بين R<sub>1</sub> و R<sub>2</sub>

إذا : (S<sub>2</sub>) = R<sub>1</sub> محتفل (S<sub>2</sub>) → R<sub>2</sub> و CC → 0

إذا : (S<sub>2</sub>) ≠ R<sub>1</sub> محتفل (S<sub>2</sub>) → R<sub>2</sub> و CC → 1 .

للمقارنة CDS تتم حل 64 بتة . وبالنسبة فإن R<sub>1</sub> و R<sub>2</sub> هما مرصفتان مزدوجتان (فترة 1.10) و S<sub>2</sub> هو عنوان كلمة مزدوجة من الذاكرة .

تُستعمل هاتين التعليقتين لتنفيذ الزمانة بين مهمتين تقسمان منطقة مشتركة من الذاكرة . عندما تتم المعادلة ، فإن كل بلوغ للعنوان S<sub>2</sub> هو ممنوع لأي مُعالج مركزي حتى نهاية عملية النقل (S<sub>2</sub>) → R<sub>2</sub> .

TM D<sub>1</sub>(B<sub>1</sub>),I<sub>2</sub> SI COP=91 TEST UNDER MASK

TM تقوم باختبار حالة البتات من البايته ذات العنوان S<sub>1</sub>.I<sub>2</sub> في قناع من 8 بتات . كل «1» ، موجود في القناع يسمح باختبار وجود بتة «1» في الموقع المناسب من البايته S<sub>1</sub> .

مثلاً : القناع 60'X أي B'01100000 يفحص وجود «1» في الموقعين 1 و 2 من البايته . ويجري إهمال المواقع الأخرى . وفي الإجمال ، فإن TM يقوم بتنفيذ عملية AND منطقية بين البايته التي تم فحصها والقناع دون تعديل البايته ولكن بتركيز كود الشرط فقط :

0 = CC : جميع البتات التي جرى اختبارها هي 0 أو القناع هو في صفر ،

1 = CC : بعض البتات هي صفر ، وأخرى هي 1 ،

2 = CC : غير مستعمل

3 = CC : جميع البتات المختبرة هي 1 .

11001110	11001110	11001110	البابة المختبرة
00110000	11001000	01011100	القناع
--00----	11--1----	-1-011--	AND
0	3	1	CC

تطبيق :

TM يبدو وكأنه ينتمي إلى CLI . وفعلاً فإن TM يُعتمد لاختبار البتات أكثر من البايتات . مثلاً ، لمعرفة ما إذا كانت البايته هي رقمية نستعمل CLI لأن القيمة يجب أن تكون محصورة بين F0 و F9 .  
TM يمكن أن تستعمل لتنفيذ تأثير متعكّد .

تمارين :

نذكر أن الدالة «AND» تسمح بجعل البتات تعادل صفرأ ، وإن الدالة «OR» تسمح بجعلها 1 وإن «EOR» تسمح بعكسها .

تمرين 1.13 . ضع في صفر ثنائي منطقة بطول  $L \geq 256$  بايته ، مرصفاً ، بايته .  
تمرين 2.13 . اكتب التعليمة التي تسمح بتركيز قيمة كود الطول في تعليمة من نوع SS .  
تمرين 3.13 . بدّل مضمّنون منطقيتين من الذاكرة ، مرصفين ، ريعين من البتات من نفس البايته .

تمرين 4.13 . تعرّف ما إذا كانت منطقة من الذاكرة مملوءة بفراغ أو بصفر ثنائي  
تمرين 5.13 . قم بإجراء تأثير يؤدي إلى تفريع مرّة على اثنتين بواسطة تحويل منطقة قناع تعليمة BC 15,... إلى BC 0,... .

تمرين 6.13 . قم بإجراء تأثير يؤدي إلى تفريع إلى جميع نقاط العبور ما عدا الأول .  
تمرين 7.13 . بدّل جميع أصفار البتات (X'F0') في عدد عشري بفراغات (X'40') .

تمرين 8.13 . البايته تسمح بتجميع حتى ثمانية مؤشرات ثنائية . لتأخذ البايته INDIC التي تجمع المؤشرات الثنائية INDEC , INDEC , INDWAIT التي تناسب على التوالي للقيم السادس عشرية X'80' ، X'40' و X'20' من INDIC (تحتل المؤشرات البتات 0 ، 1 و 2 من INDIC) . اكتب التعليقات التي تسمح :

- بتعريف INDIC ، INDEC ، INDEC ، INDWAIT ؛

- بتركيز INDWAIT في 1 ؛

- بتركيز INDEC و INDWAIT في 1 ؛

- بتركيز INDEC و INDEC في صفر ؛

- بتفريع إلى ALPHA إذا كانت INDWAIT في «1» ؛
  - بتفريع إلى BETA إذا كانت INDWAIT و INDLEC في «1» ؛
  - بتفريع إلى GAMMA عندما يكون فقط INDLEC أو INDWAIT في «1» ؛
  - بتفريع إلى DELTA عندما تكون INDWAIT و INDLEC في صفر .
- لنفترض بأننا نرغب بربط INDLEC بالبتة 7 من INDIC بدلاً من البتة 0 ، مما يتناسب مع 'X'01' بدلاً من 'X'80' . الحل الخاص بكم هل يسمح بعدم تعديل تعاملات التركيز والاختيار لـ INDLEC ؟

## 14 . عمليات الإزاحة (Shift)

### 1.14 . العمليات « المنطقية » والعمليات « الحسابية »

عند دراسة عمليات الجمع بفاصلة ثابتة ، لاحظنا ، أنه إلى جانب العمليات A ، AR و AH ، تأتي عمليات الجمع المنطقية . الفرق بين هذين النوعين من العمليات هو التالي :

- تميّز العمليات الجبرية البتة 0 ، المعتبرة كإشارة ، تجري العملية على 31 بتة مع مُرحّل عتَمَل إلى بتة الإشارة . يجري اختيار الإشارة ويمكن أن تؤدي إلى إنقطاع من نوع overflow .

- العمليات من نوع منطقي لا تأخذ بعين الاعتبار أي تمييز للبتة ذات الوزن الأكبر . تجري معالجة جميع البتات بنفس الطريقة . أي ترحيل في نهاية البتة ذات الوزن الأكبر لا يؤدي إلى انقطاع .

الإزاحة هي عبارة عن نقل إلى اليسار أو إلى اليمين لعدد « n » من المواقع لتشكيلة ثنائية موجودة في مرصف بسيط (إزاحة بسيطة) أو في مرصف مزدوج (إزاحة مزدوجة) .

عند الإزاحة تصبح البتات المطروقة . والبتات الداخلة لجهة اليمين هي دائماً صفر . أما البتات التي تدخل من اليسار فيمكن أن تكون إما « 0 » (إزاحة منطقية إلى اليمين أو إزاحة حسابية إلى اليمين لعدد إيجابي) أو « 1 » (إزاحة جبرية إلى اليمين لعدد سلمي) . سنرى السبب لاحقاً .

### 2.14 . الإزاحة الجبرية

تجري الإزاحة الجبرية على القيمة ، أي على 31 بتة (إزاحة بسيطة) أو على 63 بتة (إزاحة مزدوجة) .

- الإزاحة إلى اليمين تؤدي إلى إدخال بتات معادلة لبتة الإشارة .

- الإزاحة إلى اليسار تؤدي إلى إدخال 0. إذا جرى تعديل بته الإشارة أصبحت إنقطاع من نوع overflow بفاصلة ثابتة .

الإزاحة الجبرية تؤدي إلى تركيز كود الشرط على الشكل التالي :

CC = 0	إذا كانت النتيجة صفراً .
CC = 1	إذا كانت النتيجة سالبة .
CC = 2	إذا كانت النتيجة إيجابية
CC = 3	إذا كان يوجد overflow (تعديل في بته الإشارة في حالة إزاحة إلى اليسار) .

أمثلة :

لتبسيط العرض سنفترض إن حجم للمرصف يعادل ثمان بتات . البته ذات الوزن الأكبر هي إذاً بته الإشارة .

مرصف بسيط	مرصف مزدوج
قبل الإزاحة $\boxed{00001111}_S = +15$	$\boxed{0000000011001111}_S = +207$
بعد الإزاحة بته اليسار ثلاثة $\boxed{01111000}_S = +120$ CC = 2	$\boxed{0000011001111000} = +1656$
بعد الإزاحة بته اليمين ثلاثة $\boxed{00000001}_S = +1$ CC = 2	$\boxed{0000000000011001}_S = +25$
قبل الإزاحة $\boxed{11100101}_S = -27$	
بعد الإزاحة بته اليمين 1 $\boxed{11110010}_S = -14$ CC = 1	
بعد الإزاحة بته اليسار 4 $\boxed{01010000}_S = +80$ CC = 3 OVERFLOW	

### 3.14 . الإزاحة المنطقية

تعالج الإزاحة المنطقية 32 بتة ( إزاحة بسيطة ) أو 64 بتة ( إزاحة مزدوجة ) دون أخذ بالاعتبار البتة ذات الوزن الأكبر . البتات الداخلة هي دائماً «0» . لا يحدث إنقطاع من نوع overflow . لا يجري تعديل في CC .  
أمثلة : على ثمان بتات .

10011100

قبل الإزاحة

01110000

بعد الإزاحة لجهة اليسار 2

00100111

بعد الإزاحة لجهة اليمين 2

### 4.14 . تعليقات الإزاحة

يوجد أربع عمليات إزاحة جبرية ، أربع تعليقات لإزاحة منطقية ، وتعليمة إزاحة لعدد عشري . نشير هذه الأخيرة عند دراسة الحساب العشري .  
الإزاحة الجبرية :

SLA	$R_1, D_2(B_2)$	RS	COP=8B	SHIFT LEFT SINGLE	إزاحة بسيطة إلى اليسار
SLDA	$R_1, D_2(B_2)$	RS	COP=8F	SHIFT LEFT DOUBLE	إزاحة مزدوجة إلى اليمين
SRA	$R_1, D_2(B_2)$	RS	COP=8A	SHIFT RIGHT SINGLE	إزاحة بسيطة إلى اليمين
SRDA	$R_1, D_2(B_2)$	RS	COP=8E	SHIFT RIGHT DOUBLE	إزاحة مزدوجة إلى اليمين

### الإزاحة المنطقية

SLL	$R_1, D_2(B_2)$	RS	COP=89	SHIFT LEFT SINGLE LOGICAL	إزاحة بسيطة منطقية إلى اليسار
SLDL	$R_1, D_2(B_2)$	RS	COP=8D	SHIFT LEFT DOUBLE LOGICAL	إزاحة منطقية مزدوجة إلى اليسار

SRL	$R_1, D_2(B_2)$	RS	COP=88	SHIFT RIGHT SINGLE LOGICAL
				إزاحة بسيطة منطقية إلى اليمين
SRDL	$R_1, D_2(B_2)$	RS	COP=8C	SHIFT RIGHT DOUBLE LOGICAL
				إزاحة مزدوجة منطقية إلى اليمين

نواحد مشتركة للإزاحات المنطقية والجبرية

- تتم عمليات الإزاحة على مضمون المرصف  $R_1$ .
- بالنسبة لعمليات الإزاحة المزدوجة ، فإن  $R_1$  يجب أن يكون مرصفاً مزدوجاً حسب الاتفاق العادي (فقرة 2.10).
- التأثير الثاني  $D_2(B_2)$  ليس عنواناً :

- 1- إذا كان  $B_2$  هو المرصف 0 ، فإن البتات الست ذات الوزن الأضعف للبتة تعطي عدد المواقع المطلوب إزاحتها .  $SLA 5,3(0)$  أو  $SLA 5,3(0)$  هما عمليتا إزاحة لجهة اليسار لثلاثة مواقع ثنائية .
- 2- إذا لم يكن  $B_2$  هو المرصف 0 ، فإن المرصف المذكور يحتوي على عدد المواقع المطلوب إزاحتها . ونحصل على الإزاحة بشكل غير مباشر .  $SRDL 6,0(5)$  يزحل منطقياً المرصف المزدوج (المرصفان 6 و 7) لعدد المواقع المشار إليها في المرصف 5 .

- وحدهما عمليات الإزاحة الجبرية تقوم بتركيز كود الشرط CC حسب اتفاق الفقرة 2.14 .

تمارين :

- تمرين 1.14 - ضع في صفر مرصفاً بواسطة الإزاحة .
- تمرين 2.14 - إضرب واقسم عدداً موجوداً في مرصف على قوة لـ 2 بواسطة الإزاحة . إنقص ، بالنسبة للقسم ، اتجاه التقريب .
- تمرين 3.14 - إنقص فيها إذا كان زوج من المرصيف مزدوج / مفرد هو صفر .
- تمرين 4.14 - برمج إزاحة دائرية لمرصف بسيط .

### 1.15 . الفرز

يتعلق ذلك بترتيب جدول من الكليات التي تحتوي على أعداد بفاصلة ثابتة بترتيب تصاعدي . ولقد قمنا باختيار الخوارزم الكلاسيكي الذي يُعرف بـ « طريقة الفقاعة » . تقوم الطريقة على فحص عناصر الجدول من اليسار إلى اليمين مع تبديل العناصر المتتالية الموجودة بشكل عشوائي . نضع إلى اليمين العنصر الأكبر كما نلاحظ من المثل التالي :

5	3	2
1	5	2
1	3	5
1	3	2
1	3	5

إذا كان  $N$  هو حجم الجدول ، نبدأ العملية باعتبار الجدول الثانوي بالحجم  $N-1$  وهكذا دواليك ، طالما يوجد عملية تبديل واحدة على الأقل خلال التكرار السابق .

ولو افترضنا أنه خلال فحص الأعداد ، لم يمر أية عملية تبديل فمعنى ذلك إن الترتيب قد حصل .

البرنامج مؤلف من حلقتين BCL1 و BCL2 متداخلتين . الحلقة الداخلية BCL2 تفحص الجدول باستعمال مرصف مؤشر PTR : (PTR) هو عنوان العنصر . العناصر التي جرت مقارنتها هي إذا ((PTR)) و  $((PTR)+4)$  . يتم إنشاء الحلقة بواسطة BXL2 . المرصف المزودج INCRE/REFER . يحتوي على الزيادة 4 والحّد  $4 * (N-1) + TAB$  .

عند إجراء تبديل نقوم بتركيز البايته INDIC في 1 . الحلقة BCL1 تُكرر BCL2 طالما إن  $INDIC=1$  .

(1) نلّكس بأنه حسب الترميز لامتداد ، (PTR) يُقرأ « مضروب PTR » وهذا هو إن عبارة عن عنوان . مضروب هذا العنوان ، أي العنصر المطلوب ، يُرمز إليه بـ ((PTR)) .

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SPACE STATEMENT
103000					
103010	0000	0000	0000	1	INIT
103020	0000	0000	0000	2	POINTEUR SUR ELEMENT DE TAB
103030	0000	0000	0000	3	RECUPERER LE BLOC
103040	0000	0000	0000	4	RECUPERER LE BLOC
103050	0000	0000	0000	5	RECUPERER LE BLOC
103060	0000	0000	0000	6	RECUPERER LE BLOC
103070	0000	0000	0000	7	RECUPERER LE BLOC
103080	0000	0000	0000	8	RECUPERER LE BLOC
103090	0000	0000	0000	9	RECUPERER LE BLOC
103100	0000	0000	0000	10	RECUPERER LE BLOC
103110	0000	0000	0000	11	RECUPERER LE BLOC
103120	0000	0000	0000	12	RECUPERER LE BLOC
103130	0000	0000	0000	13	RECUPERER LE BLOC
103140	0000	0000	0000	14	RECUPERER LE BLOC
103150	0000	0000	0000	15	RECUPERER LE BLOC
103160	0000	0000	0000	16	RECUPERER LE BLOC
103170	0000	0000	0000	17	RECUPERER LE BLOC
103180	0000	0000	0000	18	RECUPERER LE BLOC
103190	0000	0000	0000	19	RECUPERER LE BLOC
103200	0000	0000	0000	20	RECUPERER LE BLOC
103210	0000	0000	0000	21	RECUPERER LE BLOC
103220	0000	0000	0000	22	RECUPERER LE BLOC
103230	0000	0000	0000	23	RECUPERER LE BLOC
103240	0000	0000	0000	24	RECUPERER LE BLOC
103250	0000	0000	0000	25	RECUPERER LE BLOC
103260	0000	0000	0000	26	RECUPERER LE BLOC
103270	0000	0000	0000	27	RECUPERER LE BLOC
103280	0000	0000	0000	28	RECUPERER LE BLOC
103290	0000	0000	0000	29	RECUPERER LE BLOC
103300	0000	0000	0000	30	RECUPERER LE BLOC
103310	0000	0000	0000	31	RECUPERER LE BLOC
103320	0000	0000	0000	32	RECUPERER LE BLOC
103330	0000	0000	0000	33	RECUPERER LE BLOC
103340	0000	0000	0000	34	RECUPERER LE BLOC
103350	0000	0000	0000	35	RECUPERER LE BLOC
103360	0000	0000	0000	36	RECUPERER LE BLOC
103370	0000	0000	0000	37	RECUPERER LE BLOC
103380	0000	0000	0000	38	RECUPERER LE BLOC
103390	0000	0000	0000	39	RECUPERER LE BLOC
103400	0000	0000	0000	40	RECUPERER LE BLOC
103410	0000	0000	0000	41	RECUPERER LE BLOC
103420	0000	0000	0000	42	RECUPERER LE BLOC
103430	0000	0000	0000	43	RECUPERER LE BLOC
103440	0000	0000	0000	44	RECUPERER LE BLOC
103450	0000	0000	0000	45	RECUPERER LE BLOC
103460	0000	0000	0000	46	RECUPERER LE BLOC
103470	0000	0000	0000	47	RECUPERER LE BLOC
103480	0000	0000	0000	48	RECUPERER LE BLOC
103490	0000	0000	0000	49	RECUPERER LE BLOC
103500	0000	0000	0000	50	RECUPERER LE BLOC
103510	0000	0000	0000	51	RECUPERER LE BLOC
103520	0000	0000	0000	52	RECUPERER LE BLOC
103530	0000	0000	0000	53	RECUPERER LE BLOC
103540	0000	0000	0000	54	RECUPERER LE BLOC
103550	0000	0000	0000	55	RECUPERER LE BLOC
103560	0000	0000	0000	56	RECUPERER LE BLOC
103570	0000	0000	0000	57	RECUPERER LE BLOC
103580	0000	0000	0000	58	RECUPERER LE BLOC
103590	0000	0000	0000	59	RECUPERER LE BLOC
103600	0000	0000	0000	60	RECUPERER LE BLOC
103610	0000	0000	0000	61	RECUPERER LE BLOC
103620	0000	0000	0000	62	RECUPERER LE BLOC
103630	0000	0000	0000	63	RECUPERER LE BLOC
103640	0000	0000	0000	64	RECUPERER LE BLOC
103650	0000	0000	0000	65	RECUPERER LE BLOC
103660	0000	0000	0000	66	RECUPERER LE BLOC
103670	0000	0000	0000	67	RECUPERER LE BLOC
103680	0000	0000	0000	68	RECUPERER LE BLOC
103690	0000	0000	0000	69	RECUPERER LE BLOC
103700	0000	0000	0000	70	RECUPERER LE BLOC
103710	0000	0000	0000	71	RECUPERER LE BLOC
103720	0000	0000	0000	72	RECUPERER LE BLOC
103730	0000	0000	0000	73	RECUPERER LE BLOC
103740	0000	0000	0000	74	RECUPERER LE BLOC
103750	0000	0000	0000	75	RECUPERER LE BLOC
103760	0000	0000	0000	76	RECUPERER LE BLOC
103770	0000	0000	0000	77	RECUPERER LE BLOC
103780	0000	0000	0000	78	RECUPERER LE BLOC
103790	0000	0000	0000	79	RECUPERER LE BLOC
103800	0000	0000	0000	80	RECUPERER LE BLOC
103810	0000	0000	0000	81	RECUPERER LE BLOC
103820	0000	0000	0000	82	RECUPERER LE BLOC
103830	0000	0000	0000	83	RECUPERER LE BLOC
103840	0000	0000	0000	84	RECUPERER LE BLOC
103850	0000	0000	0000	85	RECUPERER LE BLOC
103860	0000	0000	0000	86	RECUPERER LE BLOC
103870	0000	0000	0000	87	RECUPERER LE BLOC
103880	0000	0000	0000	88	RECUPERER LE BLOC
103890	0000	0000	0000	89	RECUPERER LE BLOC
103900	0000	0000	0000	90	RECUPERER LE BLOC
103910	0000	0000	0000	91	RECUPERER LE BLOC
103920	0000	0000	0000	92	RECUPERER LE BLOC
103930	0000	0000	0000	93	RECUPERER LE BLOC
103940	0000	0000	0000	94	RECUPERER LE BLOC
103950	0000	0000	0000	95	RECUPERER LE BLOC
103960	0000	0000	0000	96	RECUPERER LE BLOC
103970	0000	0000	0000	97	RECUPERER LE BLOC
103980	0000	0000	0000	98	RECUPERER LE BLOC
103990	0000	0000	0000	99	RECUPERER LE BLOC
104000	0000	0000	0000	100	RECUPERER LE BLOC





من الممكن أن نكتب الخوارزم على الشكل التالي :

( ← هورمز التخصيص )

```

INCR - 4
INDIC - 1
REFER - TAB+(N-1)*4
BCL1: TANT QUE INDIC ≠ 0 FAIRE
  INDIC - 0
  REFER - REFER - INCR
  BCL2:
    exploration
  FIN BCL2
FIN BCL1

```

SNAP هي عبارة عن ماكرو تعليمية نموذجية تسمح بالحصول على صورة سادس عشرية من الذاكرة . إستعمالها يتطلب فتح السجل (OPEN) ، إغلاق (CLOSE) ووصف السجل بواسطة الماكرو تعليمية PRINT NOGEN.DCB ( سطر 2 ) تسمح بإلغاء توليد كود الماكرو تعليقات .

## 2.15 . إستشارة فرقانية للجدول

يقوم البرنامج على البحث عن وجود أو غياب معلومة من داخل أحد الجداول . البحث المتسلسل يبدو صعباً ويستهلك كثيراً من الوقت عندما يصبح حجم الجدول كبيراً . من الممكن أن نستعمل طريقة الفرقان عندما تكون العناصر منظمة . والصفة هي التالية :

لنفترض جدولاً TAB من N عنصر منظم نبحث فيه عن موقع المعلومة الموجودة في MOT . نقوم باستشارة العنصر الموجود في وسط TAB ونقارنه بـ MOT . البحث ينتهي عندما نجد التعادل . وإلا نعيد الكرة ونتابع الاستشارة باختيار واحد من الجدولين الثانويين المشكّلين بواسطة القسمة السابقة حسب موقع العنصر الذي نبحث عنه بالنسبة للعنصر الوسط . بعد كل إستشارة تضيق الفسحة التي نبحث فيها إلى النصف .

سنفترض إن طول العنصر هو L وهذا الطول يعادل قوة ( أس ) P للعدد 2 ( $L=2^P$ ) . هذا سيسمح بإجراء عمليات ضرب وقسمة بواسطة الإزاحة . سنستعمل مراجع مؤشرات لبلوغ العناصر PTRINIT. سيحتوي على عنوان العنصر الأول من الجدول الثانوي ناقص PTRFIN.L سيحتوي على عنوان العنصر الأخير من الجدول الثانوي .



عدد العناصر هو إذاً :  $\frac{PTRFIN - PTRINIT}{L}$

عنوان العنصر الوسط هو :

عنوان البداية +  $\frac{1}{2} \times$  عدد العناصر  $\times L$

أي :  $PTRINIT + L + \frac{1}{2} \left( \frac{PTRFIN - PTRINIT}{L} \right) \times L$

عند القسمة على  $L$  يجب إهمال الباقي الذي قد يظهر .

البرنامج التالي جرى اختباره بعد إجراء نداء لبرنامجين ثانويين مكتوبين بلغة فورتران : ECR و LIRE . وجود نداءات بلغة فورتران من خلال برنامج رئيسي بلغة المازول يتطلب كتابة التعليقات 59 و 60 غير الموجودة إذن إلا لأسباب توافقية بإشراف النظام المستعمل (FORTRAN G, OS-VS2) .





## 16 . الحساب العشري

### 1.16 . عموميات

تقدّم التعليقات الحسابية العشرية وسائل لإجراء الحسابات على الأعداد العشرية « المتراسة packed » التي رأيناها في الفقرة 3.5.2 ج. ولاحقاً سندرس عملية تحويلها لمعطيات .

التعليقات الحسابية هي بنسق SS وتستعمل الطولين  $L_1$  و  $L_2$  للمتأثرين . يبقى طول المتأثرات محدوداً بـ 16 بايتة ( 32 رقماً عشرياً زائد الإشارة في التمثيل المتراس و 16 رقماً وإشارة في التمثيل الموسع ) لأنها تقسم المنطقة  $L$  بالنسق SS . شكل هذه التعليقات هو التالي :

COP	$L_1$	$L_2$	$B_1$	$B_2$	$D_1$	$D_2$
-----	-------	-------	-------	-------	-------	-------

ونشير إلى أنه جرت العادة بالنسبة للتعليقات SS بأن تكون القيم المؤولة في المناطق  $L$  هي بالطول المذكور في تعليمة مؤول ناقص 1 . هكذا ، فالتعليمة :

A P ALPHA (16), BETA (10)

سيتم تأويلها مع القيم الثنائية 1111 و 1001 بالنسبة للطول .

تضع التعليقات الحسابية النتيجة في المتأثر الأول الذي يتم إلغاؤه ويجب أن يكون هذا المتأثر بطول كافٍ لاستيعاب النتيجة دون حدوث overflow وقطع للعدد . يظهر overflow إذا لم يكن المتأثر الأول بالطول المناسب لاستيعاب النتيجة . عندما تكون  $L_1 < L_2$  لا يحدث overflow إذا لم يكن هناك مُرحّل (carry) خارج الإمكانات المقننة من الطول  $L_1$  . ويمكن تقنيع overflow بواسطة البتة SPM .

عند إجراء العمليات ، فإن الفاصلة لا تُمثّل والتراسف يتم لجهة اليمين ، كما يمكن حصر المتأثرات بواسطة عمليات إزاحة عشرية مناسبة .

تتحقق الدارات ، خلال التنفيذ ، من صلاحية الأرقام العشرية والإشارات .  
والنقاء عنصر غير صالح يؤدي إلى انقطاع من نوع استثناء بالمعطيات .  
التأثيرات 1 و 2 يمكن أن تندمج بشرط أن تكون بنفس المواقع (متراصة) بالنسبة  
للبيانات ذات الوزن الأضعف . من الممكن هكذا إضافة عدد إلى نفسه :  
مثلاً :

ALPHA بعنوان

0	0	0	1	2	3	4	5	6	5
---	---	---	---	---	---	---	---	---	---

التعليمة :

AP ALPHA(5), ALPHA+3(2) تجمع S123456 إلى S456

يتم تركيز كود الشوط CC حسب إشارة النتيجة .

2.16 . التعليقات

AP  $D_1(L_1, B_1), D_2(L_2, B_2)$  SS COP=FA ADD DECIMAL  
 $(S_1) + (S_2) + (S_1)$   
يتم تركيز كود الشوط CC .

ZAP  $D_1(L_1, B_1), D_2(L_2, B_2)$  SS COP=FB ZERO AND ADD  
 $(S_2) + (S_1)$   
تعادل العملية جمع عدد إلى صفر . ويتم تركيز CC .

SP  $D_1(L_1, B_1), D_2(L_2, B_2)$  SS COP=FB SUBTRACT DECIMAL  
 $(S_1) - (S_2) + (S_1)$   
تركيز CC .

MP  $D_1(L_1, B_1), D_2(L_2, B_2)$  SS COP=FC MULTIPLY DECIMAL  
 $(S_1) \times (S_2) + (S_1)$   
يجب أن نحصل على :  $8 \leq L_2$  و  $L_1 < L_2$  وإلا سيحدث انقطاع .  
CC يبقى بدون تعديل .

DP  $D_1(L_1, B_1), D_2(L_2, B_2)$  SS COP=FD DIVIDE DECIMAL  
 $(S_1) : (S_2) + (S_1)$   
يجري وضع النتيجة إلى اليسار في  $(S_1)$  . الباقي يُخزّن إلى اليمين في  $(S_1)$   
ونفس طول  $S_2$  .  
حجم نتيجة القسمة هو 8 بتات :  $L_2 - L_1$  يجب أن نحصل على  $8 \leq L_2$   
و  $L_1 < L_2$  وإلا سيحدث انقطاع <sup>(1)</sup> CC بدون تعديل .

(1) إنباء : يصلّى فلك بالطول L بلفة للأول وليس بطول القيم .

CP  $D_1(L_1, B_1), D_2(L_2, B_2)$  SS COP=F9 COMPARE DECIMAL

تجري مقارنة التآثرين ويتمّ تحليل مضبوط CC . إذا كانت أطوال التآثرات غير متعادلة ، فإن المنطقة الأصغر يجري ملؤها بصفر لجهة اليسار .

SRP  $D_1(L_1, B_1), D_2(B_2), I_1$  SS COP=F0 SHIFT AND ROUND DECIMAL (370)

يجب الانتباه إلى التآثرين الخاصين بهذه العملية . عند التحويل ، فإن  $D_2$  تأخذ الموقع الطبيعي المحفوظ لـ  $I_1$  .

-  $S_1$  هو عنوان التآثر المطلوب لإزاحته .

-  $I_1$  هو الطول .

-  $D_2(B_2)$  ليس عنواناً ؛ البتات الست ذات الوزن الأضعف والمعتبرة كعدم صحيح بالإشارة ، تدل على اتجاه وعدد الأرقام العشرية المطلوب إزاحتها . ويجري إهمال البتات الأخرى . القيمة السالبة ( مكتملة إلى 2 ) هي إزاحة إلى اليمين والنتيجة السالبة هي إزاحة إلى اليسار .

-  $I_1$  هو « عامل التحويل » يُستعمل للإزاحات إلى اليمين . تضاف قيمته إلى الرقم المستخرج بالإزاحة إلى اليمين والمُرسل المحتمل يرتد إلى اليسار .

- توضع النتيجة في  $(S_1)$  .

- لا تشترك الإشارة بعملية الإزاحة .

## 17 الصلابة بفاصلة متحركة

لم يبد لنا أساسياً شرح هذه التعليقات بكثير من العناية كما جرى بالنسبة للتعليقات السابقة . فدراسة هذه المجموعة من التعليقات لن نحمل لنا سوى قليلاً من المعلومات الجديدة حول الأولوية الأساسية لتشغيل المكتبات ، بينما نحن نهتم بالدرجة الأولى بهذه الأولوية . ولكن المستعمل الذي فهم جيداً كل ما هو سابق لن يزعج كثيراً من متابعة هذا الفصل . نفترض هنا بأن القارئ قد استوعب قراءة الفقرة 3.5.2 ب. حول الفاصلة المتحركة في تمثيل المعطيات . ولكي نتذكر بسهولة الكود الحرفي لهذه العمليات ، من الجيد أن نراجع الفقرة 2.10 المتعلقة بالترميز : الحرف النهائي «R» يختص بالتعليمة RR ، والأحرف E ، U ، D ، W و X هي نسق القصير المعايير (normalized) ، والقصير غير المعايير والطويل المعايير والطويل غير المعايير والموسع .

### 1.17 . عموميات

هذه التعليقات تعمل مع المرافف المتحركة المرقمة 0 ، 2 ، 4 و 6 بطول 64 بتة . الأعداد بفاصلة متحركة القصيرة توضع في الـ 32 بتة ذات الوزن الأكبر من المرافف خلال العمليات . في هذه الحالة فإن الأوزان الضعيفة يجري إهمالها . الأعداد الطويلة بالفاصلة المتحركة تشغل كامل المرافف والأعداد الموسعة بفاصلة متحركة تشغل مرصفين متتاليين . يجري تركيز موقع كود الشرط كالعادة :

جدول 1.17

CC	بالنسبة للتعليقات الجبرية	بالنسبة للمفاتيح
0	نتيجة صفر	متاثر 1 = متاثر 2
1	نتيجة سلبية	متاثر 1 > متاثر 2
2	نتيجة إيجابية	متاثر 1 < متاثر 2
3		

## 2.17 التعليمات

يوجد نفس الخصائص التي رأيناها لدى معالجة الأعداد بفاصلة ثابتة . في حالة الشك بالإمكان مراجعتها

LER	$R_1, R_2$	RR	COP=38	LOAD	متغيرات قصيرة
LE	$R_1, D_2(X_2, B_2)$	RX	COP=78	LOAD	متغيرات قصيرة
LDR	$R_1, R_2$	RR	COP=28	LOAD	متغيرات طويلة
LD	$R_1, D_2(X_2, B_2)$	RX	COP=68	LOAD	متغيرات طويلة

CC دون تعديل

LTER	$R_1, R_2$	RR	COP=32	LOAD AND TEST	متغيرات قصيرة
LTDR	$R_1, R_2$	RR	COP=22	LOAD AND TEST	متغيرات طويلة
LCER	$R_1, R_2$	RR	COP=33	LOAD COMPLEMENT	متغيرات قصيرة
LCDR	$R_1, R_2$	RR	COP=23	LOAD COMPLEMENT	متغيرات طويلة
LNER	$R_1, R_2$	RR	COP=31	LOAD NEGATIVE	متغيرات قصيرة
LNDR	$R_1, R_2$	RR	COP=21	LOAD NEGATIVE	متغيرات طويلة
LPER	$R_1, R_2$	RR	COP=30	LOAD POSITIVE	متغيرات قصيرة
LPDR	$R_1, R_2$	RR	COP=20	LOAD POSITIVE	متغيرات طويلة

تركيز أو تعديل CC

LRER	$R_1, R_2$ (370)	RR	COP=35	LOAD ROUNDED	المؤثر 2 الطويل
LRDR	$R_1, R_2$ (370)	RR	COP=25	LOAD ROUNDED	مجري تدويره ووضع في المؤثر الأول القصير
					المؤثر الموسع
					مجري تدويره ووضع في المؤثر الأول الطويل

دون تعديل

STE	$R_1, D_2(X_2, B_2)$	RX	COP=70	STORE	متغيرات قصيرة
STD	$R_1, D_2(X_2, B_2)$	RX	COP=60	STORE	متغيرات طويلة

CC دون تعديل

CER	$R_1, R_2$	RR	COP=39	COMPARE	متغيرات قصيرة
CE	$R_1, D_2(X_2, B_2)$	RX	COP=79	COMPARE	متغيرات قصيرة
CDR	$R_1, R_2$	RR	COP=29	COMPARE	متغيرات طويلة
CD	$R_1, D_2(X_2, B_2)$	RX	COP=69	COMPARE	متغيرات طويلة

تركيز أو تعديل CC

AER	$R_1, R_2$	RR	COP=3A	ADD NORMALIZED	متغيرات قصيرة
AE	$R_1, D_2(X_2, B_2)$	RX	COP=7A	ADD NORMALIZED	متغيرات قصيرة
ADR	$R_1, R_2$	RR	COP=2A	ADD NORMALIZED	متغيرات طويلة
AD	$R_1, D_2(X_2, B_2)$	RX	COP=6A	ADD NORMALIZED	متغيرات طويلة
AXR	$R_1, R_2$ (370)	RR	COP=3B	ADD NORMALIZED	متغيرات موسعة

تركيز أو تعديل CC

AJR $R_1, R_2$	RR	COP=3E	ADD UNNORMALIZED (op)	متغيرات قصيرة
AU $R_1, D_2(X_2, B_2)$	RX	COP=7E	ADD UNNORMALIZED (o)	متغيرات قصيرة
AWR $R_1, R_2$	RR	COP=2E	ADD UNNORMALIZED (of	متغيرات طويلة
AW $R_1, D_2(X_2, B_2)$	RX	COP=6E	ADD UNNORMALIZED (op	متغيرات طويلة

تركيز أو تعديل CC

SER $R_1, R_2$	RR	COP=3B	SUBTRACT NORMALIZED	متغيرات قصيرة
SE $R_1, D_2(X_2, B_2)$	RX	COP=7B	SUBTRACT NORMALIZED	متغيرات قصيرة
SDR $R_1, R_2$	RR	COP=2B	SUBTRACT NORMALIZED	متغيرات طويلة
SD $R_1, D_2(X_2, B_2)$	RX	COP=6B	SUBTRACT NORMALIZED	متغيرات طويلة
SXR $R_1, R_2$ (370)	RR	COP=27	SUBTRACT NORMALIZED	متغيرات موسعة

تركيز أو تعديل CC

SUR $R_1, R_2$	RR	COP=3F	SUBTRACT UNNORMALIZED	متغيرات قصيرة
SU $R_1, D_2(X_2, B_2)$	RX	COP=7F	SUBTRACT UNNORMALIZED	متغيرات قصيرة
SWR $R_1, R_2$	RR	COP=2F	SUBTRACT UNNORMALIZED	متغيرات طويلة
SW $R_1, D_2(X_2, B_2)$	RX	COP=6F	SUBTRACT UNNORMALIZED	متغيرات طويلة

تركيز أو تعديل CC

MER $R_1, R_2$	RR	COP=3C	MULTIPLY	متغيرات قصيرة ونتيجة موسعة
ME $R_1, D_2(X_2, B_2)$	RX	COP=7C	MULTIPLY	متغيرات قصيرة ونتيجة موسعة
MOR $R_1, R_2$	RR	COP=2C	MULTIPLY	متغيرات طويلة
MD $R_1, D_2(X_2, B_2)$	RX	COP=6C	MULTIPLY	متغيرات طويلة
MXDR $R_1, R_2$ (370)	RR	COP=27	MULTIPLY	متغيرات طويلة ونتيجة موسعة
MXD $R_1, D_2(X_2, B_2)$ (370)	RX	COP=67	MULTIPLY	متغيرات طويلة ونتيجة موسعة
MXR $R_1, R_2$ (370)	RR	COP=28	MULTIPLY	متغيرات موسعة

دون تعديل

DER $R_1, R_2$	RR	COP=3D	DIVIDE	متغيرات قصيرة
DE $R_1, D_2(X_2, B_2)$	RX	COP=7D	DIVIDE	متغيرات قصيرة
DDR $R_1, R_2$	RR	COP=2D	DIVIDE	متغيرات طويلة
DD $R_1, D_2(X_2, B_2)$	RX	COP=6D	DIVIDE	متغيرات طويلة

لا تتغير CC

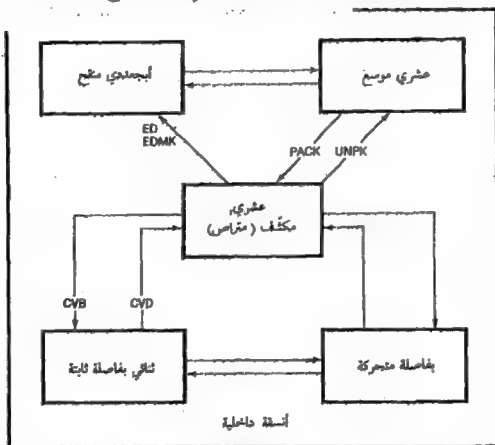
HER $R_1, R_2$	RR	COP=34	HALVE	متغيرات قصيرة
HDR $R_1, R_2$	RR	COP=24	HALVE	متغيرات طويلة

يُقسم المتأثر الثاني على 2 ويوضع نتيجة القسمة للمعالجة في المتأثر الأول .

## 18 . تعليمات التحويل والتمثيل

### 1.18 . عموميات

لقد رأينا أن النظام 370 كان يتمتع بثلاث طبقات من الدارات الحسابية العاملة بثلاث طرق مختلفة لتمثيل المعطيات الرقمية . ولكن ، المعطيات الداخلة إلى الذاكرة تكون عادةً مكونة بتمثيل أبجدي . من هنا ، فإن كل عملية حسابية على معطى رقمي داخل إلى المكتبة ، من خلال ناقل بطاقات مثلاً ، يمكن أن تتطلب عدة عمليات تحويل للتمثيل قبل معالجته بالحساب العشري ، الثنائي أو بفاصلة متحركة . المخطط 1.18 يعرض مختلف الأشكال الداخلية وعمليات النقل للمكتبة التي تتم بواسطة هذه التمايلات . الخطوط المنقطة تمثل التحويلات التي تجويزها برامج متخصصة .



خط 1.18

## 2.18 . تعليمات التحويل

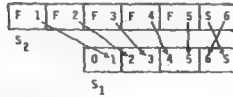
PACK  $D_1(L_1, B_1), D_2(L_2, B_2)$  SS COP=F2 PACK  
( $S_2$ )  $\rightarrow$  ( $S_1$ )

عشري مكثف عشري موسع  
(متراس)

هذه العملية تحول منطقة  $S_2$  ، يفترض إنها عشرية موسعة ، إلى عشرية متراسة . التحويل يتم من اليمين إلى اليسار بدون تحقق من صلاحية الأكواد .

إذا كانت المنطقة  $S_1$  أكبر من الضروري ، فهي تكتمل بأصفار (00) لجهة اليسار .

إذا كانت  $S_1$  قصيرة جداً يحدث قطع لجهة اليسار .  
 $S_2$  و  $S_1$  يمكن أن تتراكبا .



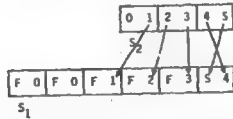
UNPK  $D_1(L_1, B_1), D_2(L_2, B_2)$  SS COP=F3 UNPACK  
( $S_2$ )  $\rightarrow$  ( $S_1$ )

عشري موسع عشري مكثف

العملية تحول منطقة  $S_2$  ، يفترض إنها عشرية متراسة ، إلى  $S_1$  عشري موسع

التحويل يتم من اليمين إلى اليسار ، بدون تحقق من صلاحية الأكواد .  
إذا كانت المنطقة  $S_1$  أصغر ، يحدث قطع أو بتر لجهة اليسار .

إذا كانت طويلة تستكمل بأصفار (F0) لجهة اليسار .  
 $S_2$  و  $S_1$  يمكن أن تتراكبا



CVB  $R_1, D_2(X_2, B_2)$

RX COP=4F CONVERT TO BINARY  
( $S_2$ )  $\rightarrow$   $R_1$

ثنائي عشري متراس

محصورة في كلمة مزدوجة

صلاحية الإشارة والبيانات الرقمية في S2 يتم التحقق منها . كل خطأ يؤدي إلى انقطاع .

يفترض بأن تكون S2 عبارة عن عنوان لكلمة مزدوجة بطول 8 بايتات . يُجَدُّ التحويل بالأعداد القصوى والصغرى التي من الممكن تمثيلها في 32 بتة ، أي :

$$-2\ 147\ 483\ 648 \leq +2\ 147\ 483\ 647.$$

CVD	$R_1, D_2(X_2, B_2)$	RX	COP=4E	CONVERT TO DECIMAL
				$R_1 \rightarrow (S_2)$
				عشري متراس
				ثنائي
				موجود في كلمة مزدوجة
				يتألف العدد العشري الحاصل من 15 رقماً إضافة إلى الإشارة : «C»
				للجمع (+) و«D» للناقص (-) . يبقى كود الشرط بدون تغيير .

### 3.18 . التنقيح والطباعة

إنَّ مضمون كلمة آلية ثنائية ، مُعطى عشري أو بفاصلة متحركة يجب ، قبل طباعته أن يخضع لتحويل معيَّن . يجب أن يتم تحويل قيمته الثنائية إلى أكواد من السيات القابلة للطباعة . قد يكون من الضروري إدخال فاصلة ، نقطة عشرية ، إشارة أو سيات تعبئة ( حالة طباعة الشيكات ) .

يوجد تعليمتان ED وEDMK تحقّقان هذا العمل بتحويل منطقة أولية ( عشري متراس ) إلى منطقة تنقيح وطباعة .  
مثلاً :

منطقة أولية

0	0	1	2	3	4	5	0
---	---	---	---	---	---	---	---

منطقة تنقيح

5	C	5	C	5	C	6	0	F	1	F	2	F	3	4	F	4	F	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

\* \* \* - 1 2 3 . 4 5

سيات تعبئة إدخال فاصلة عشرية

لكي يتم هذا ، فإن المبرمج يضع في جيّز الطباعة قناعاً مؤلفاً من :

- سمة تعبئة .

- أكواد تدلّ على : مواقع الأرقام ، المكان الذي من خلاله يتم تحويل الأصفار «0» بدون ذات معنى ، السيات المطلوب إدخالها في نهاية حقل الطباعة .

هذه التعليمات تعمل بعلاقة مع مؤشر ثنائي يُدعى « مؤشر معني » . يُوضع هذا المؤشر في «1» عندما نلتقي برقم ذي معنى في المنطقة الأولية أو عندما نلتقي مكان الأصفار التي من الواجب تحويلها .

تتعرّف هنا حل العمل الجاري بواسطة « صور » الطباعة بلغة كويول . لن يتم شرح هذه التعليمات هنا ولنصبح بمراجعة وثائق IBM370.

ED D<sub>1</sub>(L,B<sub>1</sub>),D<sub>2</sub>(B<sub>2</sub>) SS COP=DE EDIT

S<sub>1</sub> : منطقة الطباعة ، بطول L ونحتري حل القناع .  
S<sub>2</sub> : عنوان المنطقة الأولى ( للتتبع هو منطقة عشرية متفرعة ) . يتم تعديل CC حسب إشارة آخر سجل .

EDMK D<sub>1</sub>(L,B<sub>1</sub>),D<sub>2</sub>(B<sub>2</sub>) SS COP=DF EDIT AND MARK

تتتبع S<sub>2</sub> و S<sub>1</sub> بنفس الملقى . عنوان الرقم الأول في الملقى يُقرّن في المرصف 1 .  
يتم تعديل مضمون CC حسب إشارة آخر سجل .

#### 4.18 . الترجمة

TR D<sub>1</sub>(L,B<sub>1</sub>),D<sub>2</sub>(B<sub>2</sub>) SS COP=DC TRANSLATE

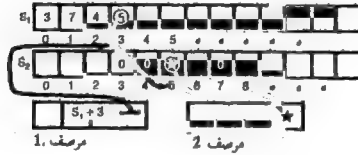
ترجمة سلسلة (S<sub>1</sub>) بطول L حسب جدول موجود في S<sub>2</sub> بطول أقصى يبلغ 256 بايتة .

قبل العملية ، فإن البايته (L)  $S_1 + X$  (  $0 < X < 256$  ) تحتوي حل الرقم p (  $0 \leq p < 256$  ) الذي يستخدم كقاعدة إدخال إلى الجدول .  
بعد العملية :  $(S_1 + X) \rightarrow (S_1 + p)$  ؛ يبقى CC بدون تعديل .

قبل العملية	S <sub>1</sub>	2	1	4															
		0	1	2	3	4	.	.	.	x	.	.	.	.	.	.	.	.	.
		جدول الترجمة																	
	S <sub>2</sub>	1	B	1		M													
		0	1	2	3	4	.	.	.	.	.	.	.	.	.	.	.	.	.
بعد العملية	S <sub>1</sub>	1	B	M															
		0	1	2	3	4	.	.	.	x	.	.	.	.	.	.	.	.	.

S<sub>1</sub> : منطقة البحث بطول L .  
S<sub>2</sub> : عنوان جدول الترجمة .  
التعليمة تستعمل المرصفين 1 و 2 .  
تؤخذ البايته الأولى من المنطقة S<sub>2</sub> بمن الاعتبار . كما في TR ، فإن قيمته الشبكية تتكفل نقطة دخول في S<sub>1</sub> .

إذا كانت البايته للنسبة  $S_2$  مختلفة عن صفر فإن قيمتها تُخزَّن في المرفص 2 وعنوان المنطقة التي تسمح بإيجاد التناسب يُخزَّن في المرفص 1 .  
 وإلا فإن العملية تتابع مع البايته التالية من  $S_1$  . يتم تركيز CC :  
 $CC = 0$  إذا كانت المنطقة  $S_1$  قد جرى إستكشافها كلياً وجميع البايتات التي جرى إستبارها من  $S_2$  كانت صفراً .  
 $CC = 1$  إذا جرى إستكشاف  $S_1$  بشكل جزئي ولم تكن البايته الأخيرة المختارة صفراً .  
 $CC = 2$  إذا جرى إستكشاف المنطقة كلياً وكانت البايته الأخيرة المختارة مختلفة عن صفر .



$CC = 1$   
 في هذا المثل ، لنستطيع متابعة إستكشاف المنطقة ، يجب إستعداد تعديل لعنوان الانطلاق والطول المستكشف .  
 $R_{exp-20}$  و  $R_{exp-7}$  يقيان دون تعديل .  
 $S_1$  لا يتم تعديلها .

## تمارين :

تمرين 1.18 - إعادة تنظيم منطقة من الذاكرة .

لنفترض منطقة ARTICLE من 10 بايتات نرغب بنقل البايتات 5 ، 6 ، 7 ، 1 ، 2 إلى المنطقة CLE

ARTICLE	A	B	C	D	E	F	G	H	I	J
CLE	F	G	H	B	C					

اكتب التعليمات التي تسمح بإجراء هذا العمل . في نفس الفكرة نرغب بعكس سلسلة من السهات . هذا النظام يستعمل لإعادة تنظيم مفاتيح الفرز .

تمرين 2.18 - لنفترض منطقة مؤلفة من 8 بايتات بقيم ثنائية موجودة بين 0 و 15 . نرغب باستبدالها بالكود EBCDIC المناسب للقيم السادس عشرية : ميجري إستبدال □ بواسطة C'0' ، و 10 بواسطة C'A' ... اكتب التعليمات المناسبة .

هذه الأوامر يمكن أن تستعمل ، بعد عملية تحويل بسيطة ، لطباعة مضمون سلاسل عشرية لكلمة من الذاكرة ، للتخضير للطباعة بواسطة DUMP (دلق) .

## 19 الانقطاع والادخال والاخراج

(Interruptions and I/O)

### 1.19 . الانقطاعات

لن يكون موضوعنا تفصيل نظام إدارة الانقطاعات هنا ، ولكن فقط إعطاء القارئ إشارات بالنسبة لطبيعة هذه المسألة . لتفصيلات أكثر ننصح بمراجعة وثائق المنشئ Principles of operation .

#### 1.1.19 . صيغة الانقطاعات

الانقطاع هو عبارة عن إشارة كهربائية ، مُرسلة من أحد أعضاء النظام ومعروفة من قِبل الوحدة المركزية . يتج الانقطاع عن حادثة تتطلب عادةً معالجة مباشرة . لبعض الحوادث صفة خاصة مستعجلة تتطلب تعليق دوران تنفيذ أحد البرامج الجارية كي يتم معالجة الإشارة المُرسلة . في النظام IBM 370 ، الحوادث القادرة على تفريم ووقف تنفيذ البرنامج قد جرى تصنيفها حسب أولوية متناقصة :

- نداء للمشرف (call supervisor) ،
- برنامج ،
- عطل في المكنة ،
- إشارة خارجية ،
- عملية إدخال - إخراج (I/O) ،
- إشارة مؤثر (operator signal) .

يرتبط بكل فئة درجة إستعجال معينة . نتكلم هنا عن ستة مستويات من الانقطاعات ونظام معالجة الحوادث يجرى حسب الأولوية المعتمدة .

### 2.1.19 . أولية الإنقطاع

نذكر بأن المفهوم الذي يدور حوله البرنامج مؤلف من كلمة حالة البرنامج PSW ومن مضمون المرافف العامة والمتحركة المرتبطة به . نشير أيضاً إلى أنه في كل لحظة ، PSW تحتوي على القيمة الحالية لعداد البرنامج . يؤدي تعليق دوران البرنامج أوتوماتيكياً

إلى تخزين مضمون هذه المراسف كي نستطيع معاودة تنفيذ هذا البرنامج المقطوع عند الحاجة . هكذا فالانقطاع يؤدي إلى إطلاق العملية التالية :

1- بشكل أوتوماتيكي ( لي بواسطة المتاد (hardware) ) ، فإن وصول إشارة الانقطاع تؤدي إلى نسخ PSW الخاصة بالبرنامج الجاري في منطقة محدة من الذاكرة ، تميز فئة الانقطاع . تدعى هذه الكلمة PSW « الكلمة القديمة » .

2- بشكل أوتوماتيكي ، يأخذ المتاد على عاتقه الكلمة الجديدة PSW الموجودة على عنوان من الذاكرة حسب فئة الانقطاع . منذ هذه اللحظة ، يمكن تنفيذ برنامج جديد . وتبدأ معالجة الانقطاع .

3- بعد الإنتهاء من معالجة الانقطاع ، يمكن معاودة العمل بالبرنامج المقطوع وذلك بواسطة إعادة ترميم الكلمة PSW وإعادة تخزين المراسف بالمعلومات التي كان يحتويها قبل قطع البرنامج .

نضيف أن معالجة الانقطاع يمكن أن تُقطع بدورها بواسطة حادثة أكبر أولوية . مجموعة البرامج التي تعالج الانقطاعات تعتبر جزءاً من نظام التشغيل وتدعى نظام إدارة الانقطاعات .

### 3.1.19 . قناع الانقطاعات

هذه الأولوية الأساسية يمكن ، ضمن بعض الشروط ، أن يتم « تقنيها » بواسطة المبرمج . بواسطة تصفير الأتعة في الكلمة PSW يمكن للمبرمج أن يمنع أخذ الحوادث الطارئة بالحسبان . هكذا يمكن إهمال الفيض overflow الناتج عن الحساب وذلك بتركيز القناع المناسب بواسطة التعليمات SPM . الإنقطاع المبرمج المقنع لا يتم أبداً ، كما يوضع الإنقطاع المقنع الناتج عن النظام في الانتظار حتى يجري رفع القناع أو القيد عنه . التعليمات SSM التي تسمح بتعديل قناع النظام هي تعليمات خاصة .

### 4.1.19 . الانقطاعات الناتجة عن البرنامج

منعطي هنا أسباب الانقطاعات الناتجة عن البرنامج . وهي تولّد عادة بسبب خطأ في البرمجة . ونجري الإشارة إليها بواسطة ظهور كود للعودة OCx يُدعى «completion code» أو كود أالنتهاء .

لتفاصيل أكثر يجب عل القارئ أن يراجع وثائق IBM الخاصة .

OPERATION EXCEPTION

code = 0C1

يتج هذا الانقطاع عندما يكون هناك محاولة لتنفيذ تعليمات بكود عملية غير صالح .

PRIVILEGED-OPERATION EXCEPTION

code = 0C2

محاولة لتنفيذ تعليمة خاصّة بينما تكون المكنة في صيغة المسألة .

EXECUTE EXCEPTION

code = 0C3

التعليمة EX تعود إلى تعليمة أخرى EX .

PROTECTION EXCEPTION

code = 0C4

يتعلّق ذلك ببلوغ موقع عممي من الذاكرة .

ADRESSING EXCEPTION

code = 0C5

يتعلّق ذلك بمحاولة بلوغ موقع غير موجود في الذاكرة .

SPECIFICATION EXCEPTION

code = 0C6

هذا الانقطاع يغطّي أكثر الحالات ، لن نذكر سوى الأكثر شيوعاً . يتعلّق ذلك بمسألة الحدود : لا تحصر التعليمة بحدود نصف كلمة أو معطى غير مسطّر كما تحتاج التعليمة التي تُرجع إليها .

DATA EXCEPTION

code = 0C7

يتعلّق ذلك بمشكلة ناتجة عن تعليمة CVB أو تعليمة عشرية .

FIXED-POINT-OVERFLOW EXCEPTION

code = 0C8

overflow في تمثيل بفاصلة ثابتة .

FIXED POINT DIVIDE EXCEPTION

code = 0C9

يتعلّق ذلك بالقسمة على صفر ، أو بنتيجة قسمة يزيد حجمها عن حجم المرصّف أو بتحويل إلى ثنائي (CVB) حيث النتيجة تزيد عن 31 بته .

DECIMAL-OVERFLOW EXCEPTION

code = 0CA

تلتقي هذه التعليمة في عملية حل أعداد عشرية ، عندما يتم فقدان البتات ذات الأوزان العليا لأن المنطقة النهائية هي أصغر من أن تحتوي على النتيجة .

DECIMAL-DIVIDE EXCEPTION

code = 0CB

يتعلّق ذلك بالقسمة على صفر في عملية بالنظام العشري .

**EXPONENT-OVERFLOW EXCEPTION**

code = 0CC

الأس الخاص بالنتيجة يزيد عن 127 والقسم العشري (mantisse) ليس صفراً .

**EXPONENT-UNDERFLOW EXCEPTION**

code = 0CD

الأس هو سلبى والقسم العشري ليس صفراً .

**SIGNIFICANCE EXCEPTION**

code = 0CE

في عملية جمع أو طرح على أعداد بفاصلة متحركة والقسم العشري هو صفر .

**FLOATING POINT-DIVIDE EXCEPTION**

code = 0CF

قسمة على صفر لأعداد بفاصلة متحركة .

**5.1.19 . تعليقات مرتبطة بالانقطاعات****SPM**  $R_1$ **RR** COP=04**SET PROGRAM MASK** $R_{1(12-7)} + CC,$ 

اتمة البرنامج



البتات من 2 إلى 7 من المرصيف العام  $R_1$  تُخزَّن (البتات 2 و 3) في CC وفي (البتان 4 و 7) قطاع البرنامج . نشير هنا إلى أن التعليمات BAL وBALR تشحن المرصيف  $R_{1(12-7)}$  بالكود CC وقناع البرنامج .

**SVC****RR** COP=0A**SUPERVISOR CALL**

هذه التعليمة تؤدي إلى انقطاع بكود I . الكلمة القديمة PSW تُخزَّن في الذاكرة على العنوان 32 والكلمة الجديدة PSW تؤخذ على العنوان 96 .

**MC**  $D_1(B_1), I_2$   
(370)**II** COP=AF**MONITOR CALL**

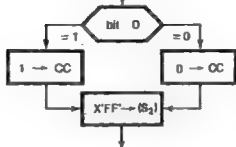
تطلق برنامج انقطاع عندما تكون بتة خاصّة من القناع للوجه في 1 .

**STCK**  $D_2(B_2)$   
(370)**S**COP=B205 **STORE CLOCK**

ة الحالية للساعة توضع في كلمة مزدوجة بعنوان  $S_2$  . البتة 31 من ساعة تزداد كل 1,048566 ثانية . ويتم تركيز كود الشرط حسب حالة الساعة .

TS D<sub>3</sub>(B<sub>2</sub>) S COP=93

TEST AND SET



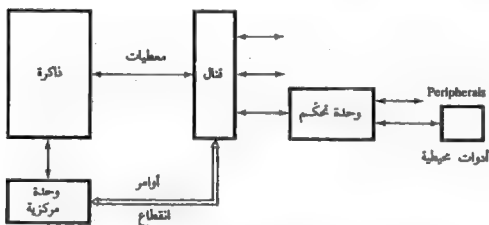
هذه التعليلة تخص البتة 0 من البتة بتوازي S<sub>0</sub> وبذلك تضع جميع البتات في 1. يتم تركيز CC. لا يمكن قطع هذه التعليلة. وتستعمل بشكل خاص للتحكم بتقاسم المصدر بين عمليتين (Processus و CROCUS) systemes des exploitation des ordinateurs, Dunod)

## 2.19 . الإدخال - الإخراج

سنعرض هنا للعمليات المهمة لإجراء المدخل والمخرج . بإمكان القارئ ، عند القيام باختباره ، إجراء إدخال - إخراج باستعمال حلقات من فورتران ، مثلاً ، أو بفضل وجود ماكرو تعليقات موجودة على النظام الذي يعمل عليه . سنعود بعد قراءة العموميات إلى دراسة ماكرو تعليقات الإدخال - الإخراج .

## 1.2.19 . تعريف وأولية الإدخال - الإخراج

عملية الإدخال - الإخراج هي عملية نقل المعطيات من الذاكرة إلى الأدوات المحيطة وبالعكس ويتم بأمر من الوحدة المركزية تحت مراقبة وتنفيذ القنال .



عند إطلاق العملية فإنها تدور دون تدخل الوحدة المركزية . يظهر القنال وكأنه مُعالج مُستقل ومُخصص لتبادل المعطيات بين الذاكرة والجهاز المحيطي . وبشكل عام ،

يوضع البرنامج الذي طلب الإدخال / الإخراج في الانتظار حتى إنتهاء عملية الإدخال / الإخراج . وهذا يعني أن تنفيذ معلق خلال مدة الإدخال / الإخراج . وهو يفقد مصادر الوحدة المركزية التي يُمكن أن تُخصَّص إلى برامج أخرى مُستَظرة التنفيذ . بعد إنتهاء عملية الإدخال - الإخراج - وهذا ما يتم إعلام النظام به بواسطة الإنقطاع - سيكون بإمكان البرنامج المقطوع أن يُعاود العمل ، وسيوضع في سجل البرامج التي تنتظر مصادر الوحدة المركزية . هنا يدخل موضوع المزامنة المفروض من الإدخال - الإخراج . يتم تأمين هذا التنظيم والإدارة بواسطة برامج (رُجل) خاصة من نظام التشغيل وهذا هو السبب الذي لأجله لا يستطيع المبرمج أن يُوجِّه بالكامل عمليات الإدخال - الإخراج الخاصة به . فهو يعطي فقط الإشارات اللازمة لنظام التشغيل ليؤمن حسن تشغيل ودوران برنامجه .

2.2.19 . المعلومات الضرورية لعملية إدخال - إخراج  
فلنفكر من خلال مثل من فورتان . لفترض عملية كتابة على الطابعة I و J هي متحولات صحيحة .

```
WRITE(6,1000) I,J
1000 FORMAT(1X,'I= ',I5,'J= ',I5)
```

إذا كانت قيمة I و J هي على التوالي -4532 و 3 ، نحصل إذاً على :

```
I=Δ-4532ΔJ=ΔΔΔΔΔ3
```

حيث Δ ترمز إلى الفسحة (البياض) الفارغة .

هذه التعليمة في الإدخال - الإخراج المستوحاة من لغة متطورة تغطي مرحلتين مختلفتين .

- لتحويل التحويلات الصحيحة I و J (ثنائي بفاصلة ثابتة) إلى سيات قابلة للطباعة .

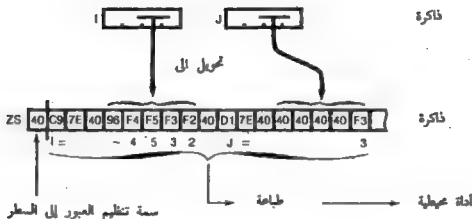
وَأَمَّن عملية الإدخال - الإخراج ، أي تبادل المعطيات .

المخطط اللاحق يُوجِّز العمليات .

النسق FORMAT يُكَمِّل إذاً القناع الذي تكلمنا عنه عند دراسة تعليقات الطابعة . المرحلة 1 تتم تحت تحكم البرنامج ، المرحلة 2 تقع على عاتق القناع .

نلاحظ إذاً أنه من الضروري معرفة :

- نوع الأداة للحيطية (رقم الوحدة المنطقية ، بلغة فورتان) ،
- العنوان ZS للمنطقة المطلوب طباعتها .



- طول ZS بالبايتات ،

- نوع الأمر ( READ أو WRITE ) .

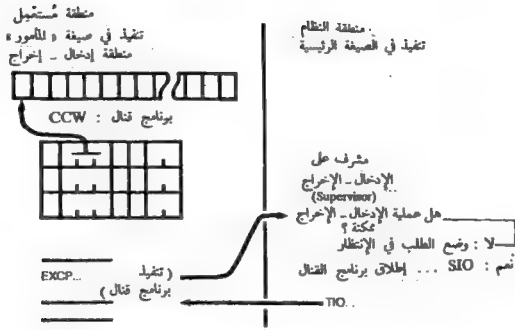
هذه المعلومات إضافة إلى معلومات أخرى ، لأن عمليات الإدخال - الإخراج هي في الواقع أكثر تعقيداً ، يتم وضعها في كلمة مزدوجة للتحكم بالقناة تدعى CCW (Channel command word : كلمة أمر للقناة) .

يلعب القناة دور الحاسب لأنه قابل للبرمجة . ستدعى «برنامج قنال» أو «برنامج وحدة تبادل» ، مجموعة الكلمات CCW المكوّنة من أوامر متتالية تتحكم بالمحيط .

الأدوات المحيطية هي عبارة عن مصادر قابلة للتقسيم والتوزيع بين عدة مستعملين . يصبح إذاً من الضروري معالجة النزاعات التي قد تولد من جراء طلبات متزامنة لنفس المصدر . لهذا السبب فإن مسؤولية إطلاق برنامج القناة تقع على عاتق نظام التشغيل الذي سيتحقق من توفر القناة والوحدة المحيطية . وبشكل آخر ، بإمكانه أن يأخذ بعض القرارات في حالة حدوث تنفيذ خاطيء لعملية الإدخال - الإخراج . الكلمة - المزدوجة ذات العنوان 40 ، بالنظام السادس عشري ، والتي تدعى CSW (Channel status word) ، تعطي بعض المعلومات حول دوران ومحاولة إطلاق الإدخال - الإخراج . المخطط الوارد حل الصفحة التالية يقوم ببعض عمليات الربط بين مختلف العناصر الضرورية للإدخال - الإخراج .

### 3.2.19 - إدخال - إخراج في المستوى المنطقي

إنّ تنفيذ عملية إدخال - إخراج بالمستوى الفيزيائي هو أمر معقد . كتابة CCW تتطلب معرفة واضحة بالمحيطات التي نعمل عليها . ونعرف أنه في أغلب الوقت تكون



عمليات الإدخال - الإخراج على المحطات البطيئة مؤجلة . عندما يقوم المستعمل بتعريف سجل طباعة ( حالة (6,...) WRITE بلغة فورتران ) ، فإن هذا السجل هو أولاً مكتوب على قرص مغناطيسي وبعد ذلك ، بواسطة برنامج خاص ، يُؤخذ لإجراء طباعة نهائية . وفي المجموع فإن رقم الوحدة المنطقي ، يُناسب أولاً فيزيائياً سجل قرص مغناطيسي وبعد ذلك سجل الطباعة . هذه العملية ، التي تحاول تبسيط إدارة المصادر المركزية والمحيطية ، تؤدي إلى زيادة الصعوبة في تنفيذ عملية الإدخال - الإخراج الفيزيائية . من جهة أخرى ، فإن تنظيم عملية إدخال - إخراج يؤدي إلى درء (Bufferization) لمناطق إدخال - إخراج . نعرف أيضاً أنه يوجد عدة تنظيمات نموذجية للسجلات وعدة طرق للبلوغ . هذه الشروط تفرض على المستعمل بأن يأمن بالكامل لنظام إدارة عمليات الإدخال - الإخراج . للقيام بذلك يجب عليه وصف المتغيرات الوسيطة المقيدة بواسطة توجيه من نوع (DATA CONTROL BLOCK) DCB . وهو سيوكل عملية الإدخال - الإخراج الخاصة به للنظام بواسطة ماكرو تعليمية خاصة (PUT GET,...) حسب نوع تنظيم السجل الخاص به . هذه الأخيرة هي موضحة في الوثائق OS/VS2 MVS (DATA Management Macro Instructions) . يقوم النظام بتوليد الكلمات CCW لنفسه ونداء المشرف الضروري . العملية الأولى للإدخال - الإخراج ستكون مسبوقة بفتح للسجل (ماكرو OPEN) والأخيرة ستكون متبوعة بإغلاق للسجل (ماكرو CLOSE) يسمح بتفريغ الدارء (Buffer) الأخير . المثال التالي يوضح ، بإشراف النظام OS ، عملية قراءة بطاقة مثقوبة وكتابة على الطباعة .

	OPEN	(CARTE,{INPUT})
	OPEN	(IMP,{OUTPUT})
	GET	CARTE,ZENTREE
	PUT	IMP,ZSORTIE
	CLOSE	CARTE
	CLOSE	IMP
CARTE	DCB	DISNAME=ENTREE,DSORG=PS,LRECL=80,BLKSIZE=400,MACRF=(GM), RECFM=FB,EODAD=SUITE
IMP	DCB	DISNAME=SORTIE,DSORG=PS,LRECL=133,BLKSIZE=665,MACRF=(PW), RECFM=FBA
ZENTREE	DS	CL'80'
ZSORTIE	OC	133C'

## 20 . الأوامر المتعلقة بالمعونة

### وتركيب البرنامج

سنقوم بجمع الأوامر ( التوجيهات ) المستعملة عند بداية ونهاية البرنامج ، التي تسمح بإعداد أعداد المواقع ، وتعريف المرافص القاعدية أو تغيير وتقطيع البرامج .

#### 1.20 . تعريف و شحن مرافص القاعدة

لقد عرفنا العنونة القاعدية ( فقرة 2.3 ) وعرضنا مثلاً على تأويل تعليمة من هذا النوع ( فقرة 3.3.6 ) من الضروري العودة الآن بشكل أكثر تفصيلاً لهذه المسألة :

إتجاهات المبرمج الأولى هي :

- 1- تحديد واحد أو عدة عناوين قاعدية .
  - 2- حجز واحد أو عدة مرافص سيتم استعمالها كمرافص قاعدية .
  - 3- شحن هذه المرافص بالعناوين المناسبة .
- النقطتان الأوليان تتعلقان بمرحلة التأويل ، والنقطة الثالثة تتعلق بمرحلة التنفيذ ولا يمكن أن نُحلّ بشكل نهائي عند التأويل لأن العنوان الفعلي لحزن البرنامج في الذاكرة لن يكون معروفاً إلا في لحظة الشحن .

#### أ - USING

هو الأمر الذي يسمح للمؤول بتحديد مرافص القاعدة وحساب الإزاحة المطلوبة لعنوان محدد رمزياً ( قاعدة ضمنية ، فقرة 2.9 ) . وشكله هو التالي :

USING Ad. base, numero des registres de base

رقم مرصف القاعدة وعنوان قاعدة USING

«Ad. Base» هو تعبير مطلق أو قابل للمثل يعتبره المؤول عنواناً قاعدياً . هذا الأمر لا يُؤلّد أية تعليمة ولذلك فهو لا يزيد من قيمة أعداد المواقع . وهو ينتمي من البرنامج المؤول .

مثلاً :

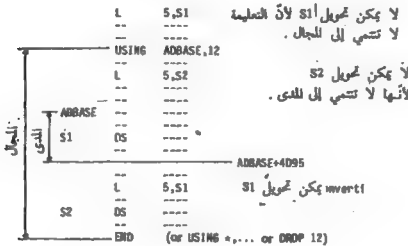
- (1) USING ADBASE,12
- (2) USING ADBASE,12,11,10
- (3) USING \*,15

الإزاحة هي كمية مكوّنة من 12 بة لا تزيد عن 4095 . وبالتالي ، فإن مدى مصرف القاعدة 12 سيمتد من ADBASE إلى ADBASE + 4095 . عندما يزيد البرنامج عن 4096 بائنة يجب استعمال الشكل (2) أو عدة أوامر USING لتحقيق العنوان . في الشكل (2) يفترض المزل أن للمصرف 12 يحتوي على القيمة ADBASE ، والمصرف 11 القيمة ADBASE+4096 والمصرف 10 القيمة ADBASE + 8192 . في الشكل (3) يفترض المزل أن العنوان القاعدي هو القيمة الحالية لعدد المواقع .

#### قواعد الإستعمال

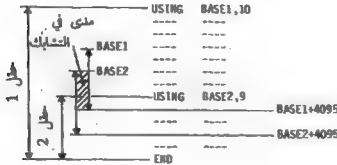
لنميز « مدى » المصرف القاعدي من الحقل المغطى بواسطة تعليمة USING . مدى للمصرف القاعدي لا يتعلّق سوى بالعنوان القاعدي المذكور في الأمر وليس بموقع USING . ويمتد من ADBASE إلى ADBASE+4095 . هذا يعني إن جميع الرموز التي تنتمي إلى المنطقة يمكن أن تعنون بناء على انتهاء التعليقات التي ترجع إليها إلى « الحقل » .

الحقل USING يمتد من الأمر ( التوجيه ) USING حتى نهاية (END) للزجلة . الأمر الآخر USING يُحدّد نفس المصرف أو يفتح الأمر DROP النهاية للحقل السابق . المثال التالي يوضح ذلك .



### حالة استعمال عدة أوامر USING

عندما يتشابه مدى عدة مرافق ، فإن الموزل يحدد بشكل جلي العناوين الرمزية المشتركة لكلا المدينين باختيار عنوان قاعدي ذلك الذي يتبع أصغر إزاحة . إذا كانت العناوين القاعدية متشابهة (BASE1 و BASE2 هي ذاتها) ، فهو يختار رقم المرصيف الأكبر . إذا كانت العناوين مختلفة ولكن المرافق متشابهة فإن الأمر الثاني USING يقطع مدى الأول



### ب- شحن مرافق القاعدة

يتوجه الأمر USING إلى مرحلة التحويل (assembling) . يجب على المبرمج أن يتوقع تعليمة تقوم ، عند التنفيذ ، بتخزين المرافق القاعدية بالعناوين الفعلية الضرورية . هذه العناوين لا يمكن أن تكون معروفة في لحظة التحويل (assembling) لأنها تتعلق بنقطة الشحن (فقرة 4.6) . المشكلة هي إذاً في كيفية معرفة طريقة استرجاع هذه العناوين . نستعمل لذلك تقنيتين : الطريقة الأولى تستعمل حالة خاصة في استعمال BALR : حيث R2 هو المرصيف 0 (فقرة 4.12) . هكذا فمن الممكن كتابة :

```
BALR 12,0
USING *,12
```

يُحزّن عنوان التعليمة BALR زائد 2 (طول التعليمة) في المرصيف 12 وهذا العنوان (●) يُحدّد كقاعدة .

الطريقة الأخرى تقوم على استعمال إتفاق عادي من النظام OS (فقرة 5.21) . يوجهه يُحزّن النظام في المرصيف 15 عنوان نقطة الدخول إلى البرنامج الذي ينتقل التحكم إليه . هذه هي طريقتنا المفضلة . سنختار كعنوان قاعدي عنوان بداية (نقطة الدخول) إلى البرنامج .

DEBUT CSECT

-----  
USING DEBUT,12  
LR 12,15

وبالتالي ، وحدها التعليقات التي لا تستعمل عنوانين رمزية يُمكن أن تظهر قبل شحن للرصيف القاعدي .

**ج - DROP**

التوجيه أو الأمر  $R_1, R_2, \dots, R_n$  DROP يُشير إلى المؤول لكي لا يستعمل للراصيف  $R_1, R_2, \dots, R_n$  كمراصيف قاعدية .

## 2.20 . تقطيع البرامج

كل برنامج مهم يجب أن يكون مقطوعاً ، أي مقسماً إلى قطع (زجل module) مستقلة . هذا ما يؤمن لنا بعض الاهتيمات : تبسيط البرامج وتقيص طول المهام ، إعطاء البرنامج كاملاً تركيبة زجلية تسمح بتسهيل عملية تعديل البرنامج ، تسهيل عمل الفريق ( العمل الجماعي )... ونحصل على ذلك بتقسيم البرنامج إلى عدة أقسام - مصدر ، باستعمال الإمكانات التي تضعها البرامج الثانوية بصرفنا ( أنظر الفصل 21 ) ، وباستعمال أوامر ( توجيهات ) التقسيم .

قسم مهم من عمل المؤول يقوم على ربط الرموز الموجودة في الزجل ( الأقسام ) بعنوانين محددة على شكل قاعدة ، مؤشر وإزاحة . ينتهي المؤول من العمل عندما يلتقي الأمر END الذي يشير إلى نهاية الزجلة . تتألف الزجلة المصدر من مجموعة من التعليقات المؤولة في مرة واحدة .

### 1.2.20 . رموز داخلية ، رموز خارجية

يمكن تصنيف الرموز التي يلتقيها المؤول في زجلة مصدريه ، في عدة طبقات .

#### 1 - الرموز المطلقة .

2 - الرموز المنقولة التي تظهر في منطقة الـوسم . وهي تسمح عادة ببلوغ تعليمة أو معطى ما . ولا يمكنها أن تظهر إلا مرة واحدة في منطقة الـوسم خوفاً من التعريف المزدوج . كما أنها داخلية ضمن زجلة المنبع ويقوم المؤول بربطها بعنوان على شكل قاعدة وإزاحة . ويقوم بتخزينها في جدول الرموز المنقولة ( المترجمة ) .

3 - الرموز التي تظهر في منطقة الـوسم ولكن من النوع « نقاط الدخول » . وتنتمي إلى زجلة المصدر ولكنها قد تكون قابلة للتسمية بواسطة أسماء من خارج هذه الزجلة . من الممكن تصنيفها في طبقتين : طبقة الرموز المستعملة . في تسمية التعليقات ، وطبقة تلك التي تستعمل لتسمية مناطق المعطيات . يقوم المؤول بتخزينها في جدول

الرموز الخارجية ESD (External Symbol Dictionary) حتى لو كانت داخلية في زجلة المصدر . رمز واحد على الأقل ينتمي إلى الفئة الأولى : الرمز الذي يشير إلى التعليمات الأولى للتنفيذ . إذا كان هذا الأمر غالباً فإن المؤول يختار كنقطة دخول عنوان التعليمات الأولى من البرنامج ويخزنه في ESD . يجب تعداد الرموز من النوع نقاط الدخول في الأمر ... ENTRY SYMB1, SYMB2, ... إذا لم تكن معتبرة كنقاط دخول إذا كانت مستعملة لتسمية القطعة ( الزجلة ) .

4 - الرموز التي تظهر في زجلة منطقة العوامل ولكن غير الموجودة في منطقة الراسم . هذه الرموز تنتمي إلى زجل مصدريه أخرى ولا يستطيع المؤول أن يربط عنواناً بها ؛ وهو يعهد بهذه المهمة إلى مُنقَح الأربطة (link editor) أو إلى الشاحن ، وذلك بتخزينها في ESD . تعتبر هذه الرموز خارجية بالنسبة لزجلة المصدر . إنها عبارة عن نقاط دخول إلى زجل أخرى وإذا فهي تنتمي إلى إحدى الطبقتين المذكورتين في 3 . ويجب أن يكون مصرحاً عنها وكأنها خارجية بواسطة الأمر EXTRN SYMB1... SYMB2... إلا إذا كانت عبارة عن أسماء برامج ثانوية مصرحاً عنها في ثابتة بعنوان من النوع V .

## 2.2.20 . أوامر التقسيم

هذه الأوامر تشير إلى بداية أو نهاية قسم من زجلة المصدر .

[ تعبير منقول ( مترجم ) ] END

يشير إلى نهاية زجلة المصدر . العنوان المناسب للتعبير المنقول يُخزن في ESD . إنه بشكل عام عنوان أول تعليمة للتنفيذ .

```

CSECT
-----
ALPHA
-----
END      ALPHA

```

يُعرف ALPHA كنقطة دخول إلى البرنامج .

قسم التحكم (Control section) هو عبارة عن قطعة منقولة من البرنامج ( قابلة للترجمة ) . هذا يعني بأنه يجب أن يربط بها مرصف قاعدة واحد على الأقل ، مما يجعل هذه الوحدة قابلة للنقل والترجمة بشكل مستقل عن باقي البرنامج . وهي تبدأ بحلود كلمة مزدوجة . يمتد قسم التحكم من بداية القسم حتى انتهاء قسم آخر .

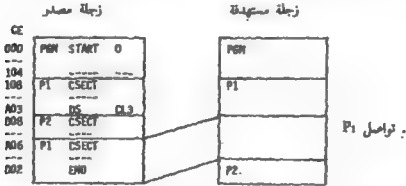
[symbole] START [constante]

[ ثابتة ] START [ رمز ]

يقوم بإعداد قسم التحكم الأول بجزلة المصدر . الثابتة الاختيارية تسمح بإعطاء قيمة أولية إلى عداد المواقع . يُخزّن الرمز في ESD .

[ Symbol ] CSBCT

يُعرف عن قسم التحكم أو يُؤشر إلى قسم داخلي . الإلتقاء الأول للرمز يشير إلى بداية القسم ، والإلتقاء التالي لنفس الرمز يشير إلى مواصلة القسم . يعمل المؤول قسماً بعد قسم : يختلف قطع القسم تكون موجودة متصلة في نفس الزجلة المستهدفة (object module) ، هكذا في المثال التالي ، يتم تأويل تواصل P1 قبل P2 . من هنا نحصل على قاعدة كيفية تطوّر CE .



تُخزّن الرموز PGM ، P1 و P2 في الجداول ESD . وهي تُمثل نقاط الدخول . تشير إلى أن جميع أقسام التحكم يجب أن تعرف بواسطة رمز ما عدا واحداً . يمكن أن يُعرف بواسطة اسم أيضاً . يجب على كل قسم ، وهذا موجود في التعريف ، أن يتمتع بحرف قاعدة . ويُعرف المؤول العنوانين الفيزيائية للقسم باستعمال هذا المرفف القاعدي الذي يجب أن يُضمن مع قيمة العنوان المناسب . يمكن لقسم التحكم أن يبدأ على الشكل التالي :

[symbol] CSBCT (أو START للأولى)  
BALR RBASE,0 (حيث RBASE هو المرفف القاعدي)  
USING \*,RBASE

سنعرض عليكم حلاً آخر لشحن المرفف القاعدي في الفصل 21 .

القسم الوهمي (dummy section) هو عبارة عن قسم مستعمل فقط لوصف المعلومات دون حجز مواقع لها في الذاكرة ويسمح إذاً بتعريف رموز دون ربطها بعنوانين في لحظة كتابة القسم الوهمي . المثال التالي سيوضح ذلك :

لنُفَرِّض البرنامج التالي الذي يستعمل المنطقتين Z1 و Z2 المنفصلتين فيزيائياً مع أنها بتركيبة متشابهة . سنقوم بتعريف التركيبة المشتركة في تركيبة وهمية تدعى ENREG وستطبقها على Z1 و Z2 عندما يصبح ذلك ضرورياً .

Z1	OS	CL80	حجز للناظر
Z2	OS	CL80	
	---	---	
	USING	ENREG,4	تعريف العنونة بالنسبة للقسم الوهمي
	I	4,=A(Z1)	
	---	---	
	---	---	تطبيق تركيبة القسم الوهمي على Z1
	L	4,=A(Z2)	
	---	---	
	---	---	تطبيق تركيبة القسم الوهمي على Z2
	---	---	
ENREG	DSECT		
NUMERO	OS	CL4	
POINTANT	OS	CL10	
NUM	OS	CL20	
ADRESSE	OS	CL46	

#### [symbole] DSECT

يُعرَّف عن بداية أو تواصل القسم الوهمي . عنونة القسم يمكن أن تتم بفضل وجود الرمز الموجود قبل DSECT أو بفضل وجود أي رمز في الوصف . يُوضع عدّاد الرموز دائماً في صفر عند بداية DSECT . يُخزّن الرمز في ESD . من هنا نلاحظ البساطة الناتجة عن هذا المفهوم . والبرجيّة ستكون مُبسّطة ومن هنا يتسج إقتصاد في استعمال الرموز .

القسم المشترك يسمح لعدة زجل مصدر ، مؤولة بشكل منفصل ولكن متّحلة فيما بينها بواسطة متّح الأربطة ، أن تتقاسم نفس منطقة التنفيذ . سنستعمل هذه المنطقة :

- لإيصال المعليّات بين زجل المصدر (فورتران ومؤوّل مثلاً) ،
  - كمنطقة عمل مؤقتة لإحدى الزجل بشرط ألاّ تستعمل في نفس الوقت .
- عند المُعالجة بالمؤوّل سيتم حجز موقع لكل زجلة ، ولكن عند المُعالجة بواسطة مُنْطَح الأربطة فإن المناطق المشتركة ستُتحد ، وفقط ستُحفظ المنطقة ذات الحجم الأكبر .

#### [ Symbol ] COM [ رمز ]

تُعرَّف عن منطقة مشتركة . يسمح النظام OS بوسم المناطق ولكن النظام DOS لا يسمح بذلك ( لا يوجد رموز ) . من الضروري ، في كل زجلة مصدر ، أن يتم

إجراء عنونة بشكل شبيه بما جرى في DSECT . يوضع عداد المراكز في صفر عند بداية القسم .

### 3.2.20 . تنقيح الأربطة (link edition)

المقررات السابقة تسمح لنا بفهم وبشكل أفضل عمل مُنقِّح الأربطة والشاحن (loader) .

مع الزجلة المستهدفة ، يقدم المزيل إلى مُنقِّح الأربطة جدولاً ESD لكل زجلة مصدر . نجد في الجدول ESD أسماء الرموز من الفئتين 3 و 4 ( فقرة 2.2.20 ) . في كل رمز نجد كود العملية من نوع الأمر المرتبط بها . إذا كان الرمز من نوع نقطة الدخول ، فإن عنوانه هو في الزجلة المشار إليها . بالنسبة للزجلة المصدر المذكورة في الفقرة 4.2.20 ، فإن الجدول ESD يكون على الشكل التالي :

EXTERNAL SYMBOL DICTIONARY

SYMBOL	TYPE	ID	ADDR	LENGTH	LDID
ALPHA	PC	0001	000000	00001C	
P1	ER	0002			
DEBUT	SD	0003	000020	00000C	
SP	ER	0004			
	ER	0005			

يكون نوع الرمز على الشكل التالي :

كود	متنص للأمر
PC	بدون وسم CSECT أو START
SD	مع وسم CSECT أو START
DM	COM
XD	DxD أو DSECT خارجي (1)
LD	ENTRY
ER	EXTRN DC V(....) أو ثابتة بعنوان
WX	WXTRN (2)

في مقابل هذه المعلومات المرتبطة بكل زجلة ، فإن مُنقِّح الأربطة يقوم بالإجابة على الطلبات الخارجية ، أي يقوم بإجراء التناسب بين الأسماء الموجودة في مختلف ESD . وإذا لم يكن بإمكان المُنقِّح أن يحل مشكلة الطلبات الخارجية بسبب جدول الزجل ESD المطلوب ربطها ، فهو يقوم بعملية بحث منتظمة في المكتبات التي يقدر على بلوغها .

(1) CXD ، DXD ، DSECT الخارجية هي غير مشروعة في هذا الكتاب .  
(2) WXTRN تقوم بهذه نفس الدور الخاص بـ EXTRN . في ما يتعلق بالساح لمُنقِّح الأربطة بالبحث الأوتوماتيكي عن الرموز بداخل المكتبة ، فإن WXTRN تمنع هذا البحث .

#### 4.2.20 . الشحن (loading)

يقوم الشحن على تخزين البرنامج في الذاكرة بدءاً من عنوان مُحدد . كما رأينا في الفقرة 2.3 ، العنوانين المتقولة لا يجب أن تتعدّل خلال هذه العملية . والأمر ليس كذلك بالنسبة لثوابت العنوان . يقوم الشاحن بخزن العنوانين الفعلية للمتأثرات المطلوبة في الذاكرة .

يجب على المؤول أن يرسل إلى الشاحن مواقع المناطق المطلوب إعادة حسابها . يستعمل لهذا الهدف RLD (Relocation Dictionary) حيث تتواجد عناوين ثوابت العنوان . الجدول ESD في المثل أعلاه هو موجود في الفقرة 3.2.20 . نذكر بأن DC V (SYMB) يحل محل :

EXTRN SYMB

DC A(SYMB)

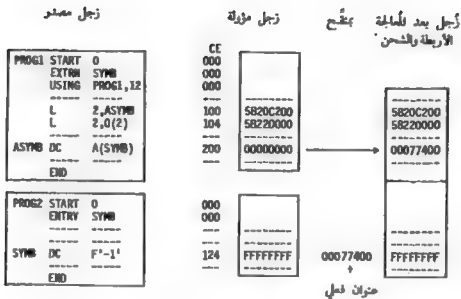
يُحفظ باستعمال ثوابت العنوان من النوع V للتعريف عن عنوان تفريع ( اسم قسم ، اسم برنامج ثانوي ... ) الرمز SYMB يُجَزَّن في ESD . ويقوم المؤول بتصغير الثابتة .

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
000000			00000	1		START 0
				2		USING 0,12
			00000	3		EXTRN ALPHA
000000	5830 C010	00010		4	DEBUT	L 3,=A(ALPHA)
000004	5850 C014	00014		5		L 5,=A(BETA)
000008	FFFFFFFF			6	ZONE	DC F=-1
000020			00020	7	PI	CSECT
				8		USING 0,11
000020	5850 C010	00010		9	BETA	L 5,=V(SPI)
000024	00000000			10	ADR	DC A(DEBUT)
000028	00000000			11		DC V(DEBUT)
				12		END
000010	00000000			13		=A(ALPHA)
000014	00000020			14		=A(BETA)
000018	00000000			15		=V(SPI)

#### RELOCATION DICTIONARY

POG-ID	REL-ID	PLAGS	ADDRESS
0001	0002	0C	000010
0001	0003	0C	000014
0001	0005	1C	000018
0003	0001	0C	000024
0003	0004	1C	000020

سنفحص في المخطط التالي كيفية تطوّر القيمة المأخوذة من قبل ثابتة عنوان من التأويل إلى الشحن :



## 5.2.20 . الاتصال بين أقسام نفس الزجلة المصدر لنأخذ المثل التالي :

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
000000				1 P1	CSECT
				2 *	USING P1,12
				3 *	
				4 *	L 3,SYMB2
000000	0000 0000	00000		5	
	*** ERROR ***				
000004	5840 C010	00010		6 *	
000008	5844 0000	00000		7	4.=A(SYMB2)
				8 *	L 4,0(4)
00000C	00000001			9 *	
				10 SYMB1	CC F*1*
000010				11 *	
				12 P2	CSECT
				13 *	
		00010		14 *	USING P2,11
				15 *	
000018	5830 C00C	0000C		16 *	L 3,SYMB1
				17 *	
00001C	FFFFFFF			18 SYMB2	OC F*-1*
				19 *	
000010	0000001C			20	END
				21	*A(SYMB2)

ولنعرض المشاكل التي يفرضها الاتصال بين قسمين عند إجراء مرحلتين من التأويل والتنفيذ .

- 1 - عند التأويل فإن أي مشكلة خاصة لن تواجهنا . يتمي القسيان إلى نفس زجلة المصدر ويمكن أن يقوم المؤول بإجراء شروط العنونة لتجميع الرموز الداخلية بشرط أن توافق القواعد المائلة إلى USING . هكذا ، فتأويل السطر الخامس لا يمكن أن يتم لأن هذه التعليمة لا تنتمي أبداً إلى حقل P2,11 USING . في القسم P1 ،

نستطيع بلوغ SYMB2 باستعمال ثابتة العنوان A(SYMB2) التي يقوم الشاحن بإعدادها بشكل مناسب . وفي المقابل ، فإن التعليمات 16 SYMB1 3, L يمكن أن تكون مؤولة .

2- عند التنفيذ ، تكون المشكلة مختلفة : التعليمات SYMB1 3, L هل مستمخ بالبلوغ إلى SYMB1 ؟

قد يسمح لنا التاويل المناسب للتعليمات بهذا الافتراض . هكذا فعلياً هذه التعليمات تسمح عند التنفيذ ، يبلوغ SYMB1 بشرط أن تكون القاعدة 12 المعنية SYMB1 تحتوي على العنوان P1 المناسب . ولكن لا شيء مؤكداً ، في مثل معاكس ، يكفي أن يكون القسم P2 مُنفذاً قبل القسم P1 كي لا تكون القاعدة 12 مشحونة بشكل مناسب . إضافة لذلك ، فإن أي مراجعة من هذه الطبيعة تناقض تعريف قسم التحكم . وبالتالي فإننا سنراجع SYMB1 في P2 بفضل وجود ثابتة العنوان .

يظهر إذاً ويوضح أن الأقسام يجب أن تُعتبر كوحلات مُستقلة في نفس الوقت الذي تكون فيه الزجل المصدرة منفصلة عند التاويل . الاتصال الرمزي بين الأقسام سيتم دائماً بواسطة ثوابت العنوان . هذه التقنية تسمح بتفادي العقبة المثارة أعلاه وتسمح بدون مشكلة بتوزيع الأقسام في مختلف زجل المصدر .  
وبإيجاز ، فإن توزيع القسم سيتم بواسطة :

$$\begin{matrix} L & R_1 = V(P1) & ( \text{أو} & A(P1) ) \\ BR & R & \end{matrix} \quad \text{حيث } R \text{ هو مصرف عام}$$

R هو مصرف عام ، بشكل عام المصرف 15 حسب إتفاقات الربط المعروضة في الفقرة 4.21 .

بلوغ الرمز يتم بواسطة :

$$\begin{matrix} L & R_1 = A(SYMB) \\ L & R, O(R) \end{matrix}$$

## 6.2.20 . ختام حول التقسيم

يعطي التقسيم وسيلة لتجزئة زجلة المصدر إلى زجل مُستقلة . عند إجراء التقسيم فإن كل شيء يجري كما لو كانت الزجل للمصدرة مترابطة .

نحرص على عدم بلوغ ، في نفس القسم ، رموز لا تنتمي إلى هذا القسم . وإذا كنا نرغب ببلوغ رموز خارجية فنستعمل الطريقة المعروضة في الفقرة 4.2.20 ، تكون إلى الشاحن مهمة إجراء الوصلة بواسطة ثوابت العنوان .

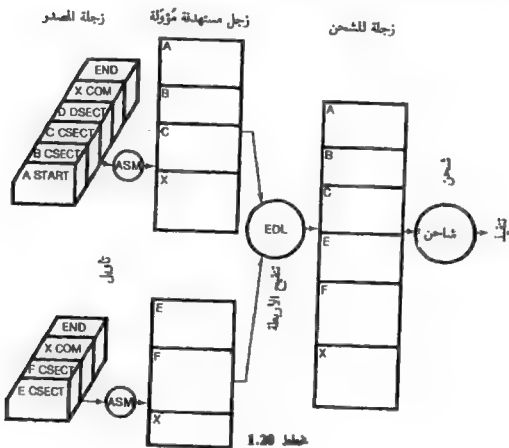
يجب على كل قسم أن يجتري على مصرف قاعلة ، ويجب شحن هذا المصرف ، في لحظة التنفيذ ، بالعنوان المناسب . ستجري دراسة هذه المسألة في الفصل التالي .

بعد أخذ هذه الاحتياطات بعين الاعتبار ، فإن التقسيم يؤدي إلى تحسين كبير في تنظيم المعالجة بالمزول . وهو يسمح ، عند الحاجة ، « بتفتيت » وبدون مشكلة البرنامج إلى زجل دون أي خوف على الترابط العام .

وبشكل عام فإن الأقسام هي برامج ثانوية . يجب إذن الاعتناء ، عند الدخول إلى قسم من هذا النوع ، بتخزين مرافق البرنامج للنادي .

ويعالج الفصل 21 هذه المشكلة . لا يجب الخلط بين القسم والبرنامج الثانوي اللذين يمثلان مفهومين مختلفين . من الممكن القول أن تقسيم البرنامج هو عبارة عن نقل قسم من العمل الجاري بواسطة المزول إلى مُنقَح الأربطة والشاحن .

سنلاحظ في المخطط التالي اختفاء DSECT من الزجلة المؤولة والموقع الوحيد المشغول بواسطة COM في الزجلة المشحونة . المكان المشغول بواسطة القسم المشترك يُعادل الحجم الأكبر بين الاثنين .



3.20 . الأوامر التي تُغيّر عدّاد المواقع  
ORG عبارة عن تعبير منقول أو مطلق . هذا الأمر يؤدي إلى تغيير الأعداد  
الطبيعي لعدّاد المواقع . وهو يسمح بشكل خاص بإجراء إعادة تعريف أو حجز مكان  
من الذاكرة . إذا كانت منطقة العناصر (القياسات) فارغة ، فإن ORG يعطي عداد  
المواقع CE القيمة التي كانت موجودة فيه عند آخر تعديل بواسطة ORG . لا يمكن أن  
يكون القياس (argument) مبلوغة في البداية .

قيمة العداد

CE

OC0	TABLE	OC	XL256*40*
OCA		ORG	TABLE+10
OD0		ORG	

LTORG عبارة عن أمر بدون قياسات . وهو يشير إلى المكان الذي يجب أن تُؤوّل  
فيه الثوابت الحرفية . في غياب هذا الأمر فإن توليها سيتم في نهاية أول قسم .  
CNOP b, w يؤدي ، بحكم علم إجراء أية عملية ، إلى زيادة قيمة عداد المواقع  
إلى الحد الأقرب لنصف كلمة ، كلمة أو كلمة مزدوجة حسب قيمتين w و b .

CNOP	0,4	بداية كلمة
CNOP	2,4	وسط كلمة
CNOP	0,8	بداية كلمة مزدوجة
CNOP	2,8	النصف كلمة الثاني من كلمة مزدوجة
CNOP	4,8	النصف كلمة الثالث من كلمة مزدوجة
CNOP	6,8	النصف كلمة الرابع من كلمة مزدوجة

4.20 . أوامر التحكم باللوائح  
ICTL يسمح بتعديل الإطار النموذجي ( الأعملة 1 ، 16 و 71 ) للتعليقات .  
ISEQ يسمح بالتحقق من الترتيب المتتالي للبطاقات .  
COPY يسمح بنسخ قسم من النص المصدر في المكتبة .  
EJECT يؤدي إلى ظهور التعليمة التالية في رأس الصفحة التالية من اللوحة .  
وهو مفيد لتوضيح نص البرنامج .  
SPACE يسمح بإدخال عدد n من الأسطر الفارغة في اللوحة .

PRINT [ON, GEN, NODATA]  
[OFF, NOGEN, DATA]

يسمح بالمحافظة على أو إلغاء اللائحة (Listing) ، توليد الماكرو تعليقات ، توليد العطاءات .

«سلسلة» TITLE يسمح بطباعة عنوان من 100 سمة في رأس كل صفحة .  
PUNCH, REPO يسمحان بتثقيب البطاقات .

5.20 . أوامر مُستعملة بإشراف النظام OS فقط  
OPSYN يسمح بتعريف مجموعة كود العمليات الخاصة المُراددة للأكواد IBM .  
هذا الأمر يمكن أن يكون مفيداً بشكل خاص لاستبدال كود - عملية خاص بماكرو عملية .

من الممكن إذاً تبديل الكود الحرفي BE ، BNE ، ... للماكرو حيث الأسماء سيصرّح عنها بشكل مرادف بسبب وجود OPSYN . هذه الماكرو تعليقات تولّد كلمة تُخزّن فيها نتيجة الاختبار الذي يسبق تعليمة التفريع بالشكل V أو F أو O أو N ، وبعد ذلك تقوم بالتفريع المناسب باستعمال التعليمة BC أو BCR . هذه السيات V أو F ستكون مرئية في العملية DUMP (دلق) وتسمح بمتابعة أثر تنفيذ البرنامج (Trace) . بالإمكان تمييز مختلف الأسماء المُولدة بواسطة SYSNDX & (فقرة 7.2.22) .

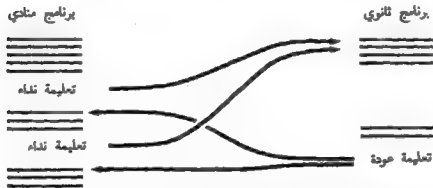
بعد مرحلة الإطلاق في العمل ، فإن إلغاء الأوامر (التوجيهات) OPSYN يؤدي إلى تفادي إدخال ماكرو التعليقات وإلبه بتثقيفها .  
من الممكن أيضاً إستعمال هذا الأمر لجعل بعض التعليقات غير عملية وذلك بجعلها مرادفة للتعليمة NOP (لا عملية) .

PUSH و POP . من الممكن عند كتابة البرنامج أن نقوم بشحن مرصف القاعدة بسرعة وأن نستعيد القاعدة القديمة لاحقاً . هذا يمكن أن يتم مثلاً ، عندما تستعمل إحدى ماكرو التعليقات قاعدة شخصية . بعد التبديل ، بواسطة المؤول ، يجري فقدان القاعدة القديمة . يسمح الأمر PUSH بتخزين المراسف وعنوان القاعدة وصيغ الأمر PRINT داخل مكدس (Stack) <sup>(1)</sup> . POP يُعاود إسترجاع المفهوم القديم بواسطة إستخراج لآخر كلمة مكدسة .

(1) للمكدس هو عبارة عن جدول مُنظّم حسب القيمة «الداخل» و«الخارج» أولاً .

## 21 البرامج الثانوية

البرنامج الثانوي هو عبارة عن سلسلة من التعليمات التي يتم تنفيذها بطلب من تعليمة نداء (Call) . عندما ينتهي تنفيذ البرنامج الثانوي يعود العمل بالبرنامج المُنَادِي. وبالتعليمة التي تتبع مباشرة تعليمة النداء . المخطط التالي يوضح هذه الألية :



كل شيء يجري كما لو كانت تعليمات البرنامج الثانوي داخلية في مكان تعليمة النداء .

بإمكاننا تقسيم البرنامج الى مهام (task) ، كل مهمة يتم حلها بواسطة برنامج ثانوي . إعداد البرنامج بكامله يصبح سهلاً ، والأقسام تصبح صغيرة . هذه الألية تسمح بتفادي إعادة كتابة التعليمات المتشابهة عندما يجب تنفيذ البرنامج في مختلف مستويات البرنامج المُنَادِي . وتطرح هذه التقنية مشكلتين :

- تخزين عنوان العودة ( العنوان الذي يتبع مباشرة عنوان تعليمة المُنَاداة ) ،
- إنتقال المتغيرات الوسيطة .

مشكلة إنتقال المتغيرات جرت إثارتها في إطار تقسيم البرنامج ولكن البرنامج الثانوي لا يُشكّل بالضرورة قسم تحكم

## 1.21 - البرنامج الثانوي وقسم التحكم

التقسيم هو عبارة عن عملية تتعلق بالتأويل ، تنجح الأربطة والشحن : أما مفهوم البرنامج الثانوي فلا يتعلق سوى بالتنفيذ . مناداة البرنامج الثانوي تؤدي ، عند التنفيذ ، إلى تعديل الدوران المتتالي للعمليات .

هكذا ، فلا شيء يعترض بأن يكون البرنامج والبرنامج الثانوي تابعين لنفس القسم . ولكن هذا النوع من التنظيم لا يُقدم جميع الفوائد التي ننتظرها من البرنامج الثانوي . فهو يربط البرنامج بالبرنامج الثانوي بينما نرغب نحن بجعل البرنامج الثانوي قابلاً للطلب والدعوة من جميع الأقسام أو الزجل . وهو لا يشكل تحسناً باتجاه تركيبة زجلية . وبالتالي لا يستعمل إلا عندما يكون البرنامج الثانوي مرتبطاً بشكل كبير منطقياً بالبرنامج المتادي .

في أغلب الأحيان يُفضل إستعمال إمكانيات التقسيم : سيشكل البرنامج الثانوي قسماً من البرنامج .. من المحتمل ، منذ لحظة تصور البرنامج الثانوي ، إستعمال هذه الزجلة في مُعالجات أخرى . يُفضل معالجة مشكلة الاتصال بين البرنامج / البرنامج الثانوي كوصلة برنامج خارجي تسمح بإمكانية تفكيك عمليات التأويل دون تعديل في الأقسام .

## 2.21 . تفريع إلى برنامج ثانوي والعودة

مناداة البرنامج الثانوي ليست سوى قطع إلزامي للدوران المتتالي للعمليات ولكن مع تخزين للعنوان التالي الذي يتبع تعليمة النادة بشكل يسمح بمعاودة العمل بالبرنامج المقطوع . تتمتع كل مكنة بلوالية خاصة للتفريع مع عودة . يستعمل النظام 360/370 التعليمتين BAL و BALR اللتين رأيتهما في الفصل 12 .

BAL R1,D2(X2,B2)  
BALR R1,R2

يكون عنوان العودة مخزناً في المرصف R1 . يكفي إذا في نهاية البرنامج الثانوي أن نشحن عداد البرنامج بالقيمة المخزنة في R1 بواسطة التعليمة BCR 15,R1 مثلاً . نحصل إذا على التركيبة التالية :

البرنامج المتادي

البرنامج الثانوي SP

-----

SP -----

L R2,=A(SP) ( أو V(SP) )  
BALR R1,R2

( تخزين المرصيف وتعريف القاعدة )

إذا كان SP خارجياً

-----

( إعادة مضمون المرافض إلى الذاكرة )  
BCR 15,R1

إذا كانت BALR موجودة على العنوان ALPHA ، فإن BCR 15,R1 تعيد تخزين ALPHA + 2 في عداد البرنامج (CO) .  
كان بإمكاننا إستعمال BAL بأحد الأشكال التالية :

- 1°) BAL R1,SP عبارة عن مرجع داخلي  
2°) L R2,A(SP) ou =V(SP)  
BAL R1,DEPLAC(R2)

الشكل الذي يسمح ، بواسطة حساب بسيط لـ DEPLAC ، بالحصول على مدخل متعددة في SP .

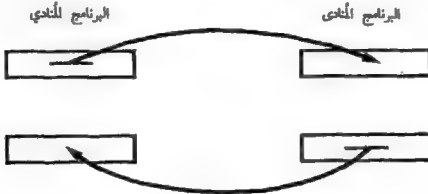
نلاحظ أنه لا يوجد فرق أساسي بين التفرعات إلى برامج ثانوية خارجية أو داخلية . وحده تعريف ثابت العنوان الخارجي هو إلزامي في الحالة الأولى .

### 3.21 . إنتقال المتغيرات الوسيطة

المشكلة الثانية في عملية الاتصال بين البرنامج والبرنامج الثانوي تكمن في عملية تبادل المعطيات . إن تقنيات عبور المتغيرات هي متعلقة ويمكن للقارىء أن يتصور الطريقة الأفضل لمسأله . ولكن من المفيد هنا أن نعرض الطرق العامة التي تساعد على الاختيار . تُستعمل اللغات المتطورة بطريقتين أساسيتين : لانتقال المتغيرات مباشرة بالقيم والانتقال بالعناوين .

#### إنتقال المتغيرات حسب القيم

ويمكن في نسخ القيمة المطلوب إرسالها إلى منطقة معروفة من البرنامج المنادي .



هذه المنطقة يمكن أن تكون خلية في الذاكرة مركزية (Local) في البرنامج المُنادى أو مرصفاً. تستعمل هذه التقنية ، مثلاً في لغة فورتران ، لاعادة قيمة إحدى الدوال إلى البرنامج المُنادي. وبشكل عام فإن النتيجة تخزن في المرصف 0 بواسطة البرنامج المُنادي .

نلاحظ إنه إذا كانت B عبارة عن متحوّلة مركزية من البرنامج المُنادى ، فإن أي تعديل في B لا يؤدي إلى أي تغيير في الخلية A .

وفي لغة المؤول ، يمكن أن نُحلّ مشكلة التبادل بالقيم بواسطة النقل بالمراسف ، حيث يُحدّد المبرمج طريقة لاستعمال المراسف .

البرنامج المُنادي

```

-----
L      1,A
L      2,B
SP      to SP

```

البرنامج المُنادى

```

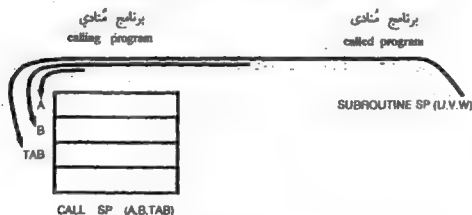
-----
ST     1,U
ST     2,V
-----

```

نشير إلى أن هذه الأوامر هي غير متوافقة مع تبادل الجداول. فعتدليّ تتطلب مكاناً كبيراً من الذاكرة . هذه الطريقة هي غير مناسبة إلا عندما يكون عدد المعطيات المطلوب إرسالها قليلاً .

انتقال المتغيرات بواسطة العناوين

وتكمن هذه الطريقة بإرسال عناوين المتغيرات إلى البرنامج المُنادى . يعمل البرنامج المُنادى إذاً على معطيات البرنامج المُنادي . يبلغ البرنامج المُنادي قيم المتغيرات بواسطة العنوان غير المباشرة . أي تعديل ، في البرنامج المُنادي ، في قيمة منقولة ، معناه تعديل منطقة من البرنامج المُنادي . هذه الطريقة هي نفسها المستعملة للإرسال بواسطة CALL (Call SP name, arguments list) في فورتران . المخطط التالي يوضح لنا عملياً كيف أن متحوّلات البرنامج المُنادي تصبح مركزية في البرنامج المُنادي .



تُدهى متغيرات وهمية الرموز A ، B ، TAB الواردة في تعليمة النداء لأنها تتمتع فعلياً بقيمة معينة في لحظة النداء أو عند العودة .

تُدهى متغيرات شكلية الرموز U ، V ، W من SP التي ليست سوى أسماء تمثّل ، في لحظة النداء ، الرموز A ، B ، TAB من البرنامج التالي .

في لغة اللؤلؤ بإمكان المبرمج تصوّر عدة حلول لنقل المتغيرات إلى البرنامج المركزي ، فلنذكر البعض منها .

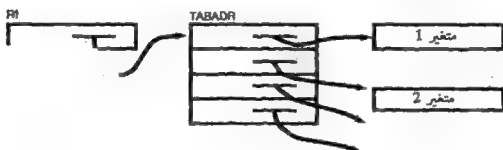
1 - نضع المتغيرات في الجدول TAB ونرسل عنوان الجدول بواسطة أحد المرافيف .

نداء		برنامج ثانوي
L	R1,=A(TAB)	يتم بلوغ المتغير □ بواسطة
L	R15,=A(SP)	L R4,DEPLAC(R1)
BALR	R14,R15	أو بالتأشير
		L R4,0(R5,R1)
		وعندئذ يوضع المتغير بتصرفه في R4

2 - نضع عنوان الجدول TAB مباشرة بعد تعليمة النداء

النداء		برنامج ثانوي
L	R15,=A(SP)	R14 يسمح ببلوغ TAB .
CNOF	2,4 ( تراصف )	العودة تتم بواسطة :
BALR	R14,R15	BC 15,4(R14)
DC	A(TAB)	

3 - تكون المتغيرات عادة غير متراصة في البرنامج ونُفضّل عادة اعتماد التقنية المستعملة بواسطة المصنّفات . نقوم بإرسال عنوان الجدول الذي يحتوي على عناوين المتغيرات بواسطة أحد المرافيف .



برنامج ثانوي		برنامج ثانوي	
L	R1,=A(TABADR)	WORK EQU ...	مرصف عمل
L	R15,=A(SP)		
BALR	R14,R15	L	WORK,0,(R1)
		L	WORK,0,(WORK)
			WORK الأول في
		L	WORK,4,(R1)
		L	WORK,0,(WORK)
			WORK الثاني في

هذا الحل هو المعتمد في لغة فورتران ، ويسمح ، في لغة المؤلف ، باستعادة المتغيرات المرسله بواسطة أحد البرامج فورتران وبالعكس .

نشير هنا إلى الفرق بين المتغيرات للرسلة ومتغيرات العودة ، وهي تنتمي إلى البرنامج الثانوي . كما تفصل إستعمال مرصاف حسب نفس الاتفاقات المستعملة في أنظمة التشغيل ( فقرة 4.21 ) . تسمح التعليمات CALL بإرسال من هذا النوع .

#### 4.21 . إتفاقات الإتصال بين النظام والبرنامج

يبدأ التنفيذ منذ اللحظة التي يتم فيها إعداد عدّاد البرنامج وتخزين عنوان التعليمات الأولى للتنفيذ فيه . يقوم نظام التشغيل بهذه المهمة ، مما يفترض علينا اعتبار كل برنامج مستعمل كبرنامج ثانوي للنظام . من هنا فإن برنامج المستعمل يجب أن يبدأ بتمهيد يتعلق بشروط إستعمال المرصاف من قبل النظام .

تسمى المرصاف 0 ، 1 ، 13 ، 14 و 15 مرصاف ربط «linkage registers» في وثائق المصنّع . وتستعمل بواسطة النظام والمصرفات بشكل نموذجي وهذا هو السبب الذي من أجله يعتمد المستعمل على نفس الاتفاقات في الاتصالات مع البرامج الثانوية الخاصة به . في النظام OS ، يجب على البرنامج الثانوي أن يجمع مرصاف المنادي في منطقة تدعى SAVE AREA ، تنتمي إلى البرنامج المنادي . تجلّد تركيبة هذه المنطقة على الشكل التالي :

#### الكلمة المحتوى

- 1 تستعمل بواسطة اللغة PL/1
- 2 عنوان SAVE AREA الداخلي السابقة (الخاصة بالمنادي) .
- 3 عنوان SAVE AREA التالية (الخاصة بالمنادي) .
- 4 عنوان العودة إلى المنادي (مرصف 14) .
- 5 عنوان نقطة الدخول إلى البرنامج (مرصف 15) .
- 6 مرصف 0 .
- 7 مرصف 1 .

.....  
18 مرصف 12 .

عندما ينقل النظام التحكم إلى البرنامج :

- يحتوي المرصف 15 على عنوان نقطة الدخول إلى البرنامج . بإمكان البرنامج المتأدي أن يشحن المرصف القاعدي الخاص به بواسطة التعليمة 15, LR REGBASE , باعتبار نقطة الدخول وكأنها عنوان قاعدي .

- المرصف 14 يحتوي على عنوان العودة .

- المرصف 13 يحتوي على العنوان بـ SAVEAREA للبرنامج المتأدي . نجد هنا شرح استعمال القاعدة 13 في التعليمة 13(12,12,14 STM الموجودة في جميع التمهيدات للبرامج .

- المرصف 1 يحتوي على عنوان جدول الكلمات التي تحتوي على عناوين المتغيرات الوهمية المتوقعة . هذا الإتفاق يستعمل ، مثلاً ، عندما يطلب برنامج فورتران برنامجاً آخر بلغة الموزل .

- المرصف 0 ، يستعمل ، عند العودة ، لإرسال نتيجة إحدى الدوال ( مثلاً الدالة FUNCTION في فورتران ) .

وبالنتيجة ، ومنذ اللحظة التي يأخذ فيها البرنامج المتأدي التحكم ويعود إلى التنفيذ ، فإنه :

- يُعرّف المنطقة الخاصة به بـ SAVE AREA ،

- يُخزن مرادف البرنامج المتأدي بواسطة :

STM 14, 12, 12 (13)

في المنطقة SAVE AREA للمتأدي

- يعرف مرصف قاعدة ويشحن فيه قيمة معينة بواسطة :

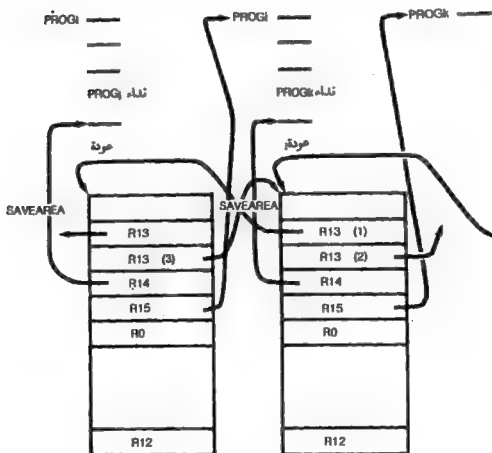
15, LR ...., 0 أو BALR .....

- يقوم بإجراء الوصلة بين المناطق SAVE AREA : ويُخزن ، في الكلمة الثانية من المنطقة SAVE AREA الخاصة به عنوان المنطقة الخاصة بالبرنامج المتأدي ( مرصف 13 ) وفي الكلمة الثالثة من المنطقة SAVE AREA الخاصة بالمتأدي ، عنوان المنطقة SAVE AREA الخاصة به .

عند العودة ، فإن البرنامج المتأدي يعيد تخزين مرادف البرنامج المتأدي مما يؤدي إلى العودة بواسطة 14 BR .

بإمكانه استعمال المرصف 15 لترميم كود العودة .

المخطط التالي يوضح عملية الربط بين المناطق SAVE AREA .



مخطط 1.21

ملاحظات : إذا كان البرنامج المتادي ، PROG<sub>i</sub> مثلاً ، لا يتقل التحكم إلى برامج ثانوية أخرى كالبرنامج PROG<sub>k</sub> ، فلا حاجة لتعريف SAVE AREA لهذا البرنامج . من الواجب إذا السهر على حماية المرصف 13 الذي يسمح بإعادة مفهوم التنفيذ الى البرنامج المتادي .

(1) يتصلق ذلك بالمرصف R13 من PROG<sub>i</sub>

(2) يتصلق ذلك بالمرصف R13 من PROG<sub>k</sub>

(3) يتصلق ذلك بالمرصف R13 من PROG<sub>i</sub>

L.O.C	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
72000				1	PROGJ	CSECT ON
72070				2	PROLOGUE	SAUVEGARDE DES REGISTRES DE L'APPELANT
72070				3		SYNCHRONISATION 14.12.12(13)
72070	90EC D00C	0000C		4		DEFINITION ET CHANGEMENT DU REGISTRE DE BASE
72070	18CF		00030	5		USING PROGJ.12
72070	5000 C01C	0001C		6		SAUVEGARDE DE R13 DANS LA SAVE AREA DE CE PROGRAMME.
72070	182D C018	00018		7		SAUVEGARDE DE L'ADRESSE EN LA SA DE L'APPELANT
72070	41D3 C018	00018		8		LR 2.13 SAVEAREA
72070	5002 C008	00008		9		ST 13.812
72070	47F9 C008	00008		10		B DEBUT
72070	00018			11		DEFINITION DE LA SAVE AREA
72070	00018			12		OS 18F
72070	00018			13		DS ON
72070	00018			14		DEBUT
72070	00070	00070		15		SEQUENCE D'APPEL DE PROSK
72070	00070	00070		16		CALL 14.13
72070	00070	00070		17		CALL 14.13 (PROSK)
72070	00070	00070		18		END
72070	00070	00070		19		ON DE RETOUR VERS PROGJ
72070	00070	00070		20		SEQUENCE 13.812
72070	00070	00070		21		LR 2.13 SAVEAREA+4
72070	00070	00070		22		ST 14.12.12(13)
72070	00070	00070		23		BR 14
72070	00070	00070		24		END
72070	00070	00070		25		SV (PROSK)

# EXTERNAL SYMBOL DICTIONARY

SYMBOL	TYPE	TO	ADDR	LENGTH	LOAD
PROGJ	SD	0021	7003C0	00070	
PROSK	SV	0021	7003C0	00070	

التعليمة STM تسمح بترتيب مرادف متتالية عند كل رغبة باستعمال مرادف متجاورة .

إتفاقات الربط المعرفة سابقاً تسمح بطلبات المتادة الداخلة ضمن البرامج . وهي لا تسمح أبداً بإجراء طلبات متادة تكرارية تحتاج إلى تعريف مكدس (STACK) خزن للنص . هذه الأوليات ليست موضوع هذا الكتاب . ولكن نشير إلى أن النظام OS يضع بتصرف المستخدم الوسائل لتعريف وإدارة منطقة من الذاكرة لكتابة برامج تكرارية ( ماكرو GETMAIN ) .

وللمحاجة إلى التماسق والتوافق ، فإن المبرمج سيقوم بنفس عمليات الاختيار كالنظام OS في استعمال المرادف لإجراء الوصلات بين البرامج الثانوية .

## 22 . التآويل المشروط وماكرو التعليمات

### 1.22 . التآويل المشروط

التآويل المشروط هو عبارة عن خطوة جديدة في التطور من لغة المكنة إلى اللغة المتطورة . ويتعلق ذلك بلغة تسمح بإنشاء وتوليد ، في مرحلة ما قبل التآويل ، نص مستهدف (object text) يمكن معالجته بواسطة المؤول . النص المؤول الناتج يمكن ، حسب القيم الأولية المخصصة لتحولات التآويل المشروط ، أن يتغير من تآويل إلى آخر . بإمكاننا مثلاً ، إدخال ، خلال مرحلة إعداد البرنامج ، متتالية من التعليمات ( طباعة وسيطة تسمح بمتابعة أثر (trace) البرنامج) التي ، بواسطة تعديل بسيط للقيم الأولية لتحولات التآويل المشروط ، سيتم إلغاؤها عند التآويل النهائي . هذه العملية ، مضافة إلى استعمال الماكرو تعليمات<sup>(1)</sup> تجعل المؤول قريباً من اللغة المتطورة ، وتسمح للمبرمج بأن يجهز بوسائل كالتعليمات : DO ... WHILE ، ... PERFORM التي تُسهّل البرمجة .

من غير الممكن هنا عرض جميع إمكانيات التآويل المشروط . سنحاول عرض الخطوط العريضة لهذه الطريقة بواسطة أمثلة توضح لنا العملية .

### 1.1.22 . متحولات وثوابت التآويل المشروط

التآويل المشروط يُعالج رموزاً بقيم قابلة للتغير : وهي عبارة عن متحولات التآويل . تبدأ أسماؤها بالرمز « & » ، وتحتوي على أكثر من ثمان سمات أبجدية ، بما فيها « & » . السمة الثانية يجب أن تكون حرفاً . متحولات التآويل هي من ثلاثة أنواع A ، B و C أي حاسوبية ، منطقية وأبجدية . يمكنها أن تكون مركزية بداخل ماكرو - إجراء والكود - المفتوح<sup>(2)</sup> (Open-code) أو شاملة (كلية) في جميع ماكرو - الإجراءات وفي الكود المقترح . يجب أن يصرح عن جميع متحولات التآويل ، المركزية

(1) مُصطلح معرف في 2.6

(2) الكود المفتوح (Open code) : قسم من كود المصدر يكون موجوداً خارج ويدل الماكرو .. تعريفات

والكَلْيَة ، قبل إستعمالها . ويتم التصريح حسب نوع التحوّلة A ، B أو C :

( مركزية ) LCLA ... LCLB ... LCLC ...  
( كَلْيَة ) GBLA ... GBLB ... GBLC ...

لا يجب أن يُصرّح عن متغيرات الماكرو-تصريف ( فقرة 1.2.22 ) . عند التصريح  
توضع التحوّلات A و B في صفر ويجري إعداد التحوّلة من النوع C في « سلسلة فارغة  
من السّيات » .

لا يمكن بلوغ متحوّلة مُصرّح عنها عل أنها مركزية إلا في نفس الماكرو-تعريف وفي  
الكود المفتوح . أما التحوّلة المُصرّح عنها « شاملة » ( كَلْيَة ) ، فيمكن بلوغها من مأكرو  
تعريفات أخرى .

يمكن أن تكون متحوّلة التأويل للمشروط عبارة من متحوّلة مؤشّرة ، وفي هذه  
الحالة يجب أن يتم التصريح عنها في مستوى LCLx أو GBLx ، كما يُصرّح عن الجدول  
في فورتران . هكذا فإن :

LCLA &TAB(20)

تصرّح عن &TAB كجدول من 20 عنصراً نستطيع بلوغه بواسطة أحد  
الأشكال التالية :

&TAB (تمير حسابي)      مثلاً : &TAB(&I+3)  
&TAB (متحوّلة مؤشّرة)      مثلاً : &TAB(&VAR(&I))

التعير الذي يعطي قيمة المؤشر يجب أن يكون إيجابياً وأن لا يزيد عن حجم  
الجدول المشار إليه في التصريح .

الثوابت الحسابية عبارة عن أعداد صحيحة بإشارة أو بدون إشارة حيث يجب أن  
تكون قيمتها بين : (  $2^{31} - 1$  و  $2^{31}$  ) .

تأخذ الثوابت المنطقية القيمة 0 أو 1 التي تناسب الغلط والصحيح . الثوابت من  
النوع سلسلة سيات تحتوي على عدد من 0 إلى 255 سمة محصورة بداخل فواصل عليا ،  
ويمكن أن تكون مؤشّرة .  
أمثلة :

(4) 'ABCDEF' ، تعادل 'D' ،

(2,3) 'ABCDEF' تعادل 'BCD' ،

المؤشر الأول يعطي الموقع الأولي للسلسلة الثانوية والثاني يعطي طولها .

### 2.1.22 . أسماء الأوسمة

منطقة الرمز من أمر تأويل مشروط يمكن أن تحتوي على وسم تأويل مشروط . إنه عبارة عن رمز يبدأ بالنقطة . و يسمح ببلوغ أمر تأويل مشروط . لأسماء الوسم مدى مركزي .

### 3.1.22 . أوامر التخصيص SET:

تقوم بتخصيص قيمة معينة إلى متحولة التأويل المشروط ، تتعلق بنوع المتحولات A ، B ، C وتتم بواسطة SETA ، SETB أو SETC . تشير إلى أن متحولة التأويل التي تحصل على التخصيص موجودة في المنطقة المحجوزة عادة للوسم . ولو افترضنا أن A ، B و C هي متحولات من النوع A ، B و C . نكتب :

منطقة الرمز	منطقة العملية	منطقة المعامل
BA	SETA	تعبير حسابي
BB	SETB	(تعبير منطقي)
BC	SETC	'تعبير أبجدي'

وبشكل عام ، بحسب التعبير ونحزّن القيمة الناتجة في متحولة التأويل الموجودة بلجهة اليسار .

### التعابير الحسابية

وتكتب بواسطة المؤثرات + ، - ، \* و / (قسمة صحيحة بدون باق) . التقييم يتم من اليسار إلى اليمين بقواعد الأولوية العادية .  
أمثلة :

القيمة التي نأخذها المتحولة

BA1	SETA	10	10
BA1	SETA	BA1+1	11

### التعابير المنطقية

تكتب بداخل أهلة بواسطة المؤثرات NOT ، AND و OR المذكورة في الترتيب التناقصي للأولويات . ويفضل وجود مؤثرات العلاقة بإمكاننا إجراء المقارنات بين التعابير الحسابية .

GT	GE	NE	EQ	LE	LT	مؤثرات علاقة :
>	≥	≠	=	≤	<	المعنى

يجب أن تكون المؤثرات محاطة بفراغات .

أمثلة :

804 SETB (801 OR 802 AND 803)  
805 SETB (8A1 GT 8A2)  
806 SETB ('8C' EQ 'ALLOC')

تعايير من نوع سلسلة سيات

هي عبارة عن مجموعات من الثوابت والمتحولات من النوع الابعدي المحصورة بداخل فواصل عليا . المؤثر « . » ( نقطة ) يسمح بإجراء عمليات الاتحاد<sup>(1)</sup> . الترميز المؤشر يسمح باستخراج السلاسل الثانوية .  
أمثلة :

		القيمة التي تأخذها المتغيرة	
8C1	SETC 'CHA'	CHA	
8C2	SETC '8C1'	CHA	
8C3	SETC '8C1'. 'INE'	CHAI NE	
	ou '8C1. INE'		
8C4	SETC 'CHAI NE' (2,5)	HAINE	
	الطول : 4 الترتيب : 1		
8C5	SETC '8C4' (1,3). '8C4' (5,2)	HAIE	
8C6	SETC 'L' 'NON'	L'NON	
8C7	SETC '5'	5 (caractère)	
8C8	SETC '8C7..25'	5.25 (un seul point)	
8C9	SETC '8A+10'	si 8A = 10 alors	
	ou '8A.+10'	10+10 et non 20	
8C10	SETC '8C18C1'	CHACHA	
	ou '8C1.8C1'		

نشير (8C10) إلى أن النقطة في عملية الاتحاد هي إختيارية عندما نجمع بين متحولتين من السيات لأن الفاصل & لا يسمح بقيام أي نوع من الإجماع .

عندما ندخل المتحولات من النوع A إلى يمين الأمر (8C9) SETC ، فإن قيمة المتحولات تستبدل بالمتحولات ولكن بدون إجراء أية عملية .

التعايير من النوع سلاسل السيات هي مهمة لأنها تسمح بإنشاء رموز أو بناء تعليقات لإتحاد متتالية . هنالك أمثلة توضح إستعمالها عند دراسة الماكرو - إجراءات .

(1) عملية الربط - جمع سلسلتين ABCD و EF منه تشكيل السلسلة ABCDEF .

4.1.22 . أوامر التفريع إلى أوصمة التأويل  
التفريع الإلزامي يتم بواسطة AGO والتفريع المشروط بواسطة AIF . ويكتبان :

وسم للتأويل المشروط      AGO      [ رسم التأويل المشروط ]  
وسم تأويل مشروط ( تمييز منطقي ) AIF      [ رسم تأويل مشروط ]

أمثلة :

إذهب إلى SUITE  
إذا &C تعادل OUI ( نعم )  
إذهب إلى ET1 ، وإلا تابع بالتالي .  
AGO .SUITE  
AIF . ('&C' EQ 'OUI').ET1

#### 5.1.22 . الأمر ANOP

هو أمر « بدون عملية » يسمح بتعريف رسم معين (Label) . ويُستعمل بشكل خاص عندما نرغب بإجراء تفريع إلى أمر ( توجيه ) SETx ، ويكون حقل الوسم العادي مشغولاً بمتحولة .

	AGO	.SUITE
	---	-----
.SUITE	ANOP	
AVAR	SETA	AVAR+1

6.1.22 . أمثلة على استعمال التأويل المشروط  
سنذكر عدة أمثلة عند دراسة ماكرو - الإجراءات . هنا نكتفي بتفصيل بعض النقاط

#### مثل 1

نرغب ، خلال تنفيذ البرنامج ، بإجراء تأويل مجموعة من التعليمات ( طباعة وبسيطة مثلاً ) بإلغاء تعليمات التأويل النهائية دون سحب البطاقات المناسبة لها . سنخضع إذاً تأويل هذه التعليمات للقيمة التي تأخذها متحولة التأويل التي تدعى هنا & TEST

&TEST	SETA	1	( مرحلة البدء بالعمل )
---	---	---	
	AIF	(&TEST EQ 0).SAUT	
	---	---	تعليمات للتأويل
	---	---	خلال مدة الاختبار
.SAUT	---	---	

بجعل التحولة TEST & تعادل صفراً تكون قد ألفينا تأويل هذه التعليمات .

مثل 2

إنشاء نصّ معين .

التأويل المشروط يمكن أن يستعمل لإنشاء نصّ متحوّل من تأويل إلى آخر . يمكن لهذا النصّ أن يكون رمزاً أو تعليمة .

يؤدي إلى توليد الأمر :  

RAND	EQU	AND
R1	EQU	1

 إذا كانت التحولة AND تعادل 1

## 2.22 . الماكرو - إجراءات

باستعمال الماكرو إجراءات نحدد أولية التأويل المشروط فالتالي :

الماكرو إجراء هو عبارة عن برنامج يعمل إسمياً مؤلفاً من سلسلة من التعليمات وأوامر التأويل المشروطة وغير المخصصة بالأوامر MACRO و MEND .  
 مثلاً : الماكرو تعريف التالي :

لائحة المتغيرات الشكلية إسم

MACRO	
SOMME	AV, BV, BV
{	
L	1,AV
A	1,BV
ST	1,BV
MEND	

سطر توافقي      جسم      الإجراء

سيكون الماكرو تعريف موجوداً خارج البرنامج (open code) الذي يُراجعه .  
 بإمكان الماكرو تعريف أن يكون موجوداً في مكتبة -المستعمل أو مكتبة المؤول .  
 الماكرو تعليمات هي إذا السطر من البرنامج الذي يطلب من المؤول إدخال نصّ النموذج في البرنامج باستبدال المتغيرات الشكلية بالمتغيرات الفعلية .  
 مثلاً :

يؤيد التالية :  

SOMME	A,B,C
L	1,A
A	1,B
ST	1,C

 المتغيرات الوسيطة الفعلية

نفترض عندئذٍ بأن هذا النظام ، المزود بالتأويل المشروط ، يسمح بإنشاء نماذج ستاندارد لبرامج يقوم المؤول بجعلها متوافقة مع كل حالة خاصة حسب قيم محاولات التأويل المشروط .

## 1.2.22 . تتقبل المتغيرات

كما في حالة البرامج الثانوية ، المتغيرات الشكلية هي متغيرات السطر النموذجي في الماكرو تعريف والمتغيرات الفعلية هي متغيرات الماكرو تعليمة . المتغيرات الشكلية هي رموز تسبقها السمة « & » .

يتكوّن السطر النموذجي في الماكرو تعريف على الشكل التالي :

اسم الإجراء	لائحة المتغيرات الشكلية
PROC	AU,BNO=3,&QTE=,AV,&RES=5,&M,&X

قيم نحو النقصان ( 0 أو سلسلة فارغة إن لم يجر تحديدها ) .

المتغيرات الشكلية هي على نوعين :

- متغيرات الوضع : &X و &W , &V , &U في المثل ،
- متغيرات الكلمة المفتاح : &NO ، &QTE و &RES . ويميّزها بكون أسماؤها متبوعة بالرمز « = » ورّبّا بالقيمة التي تأخذها نحو النقصان ، قيمة تساوي « السلسلة الفارغة » في حال عدم تحديدها . ويتكوّن سطر نداء الماكرو تعريف كما يلي :

اسم الإجراء	لائحة للمتغيرات الفعلية
PROC.	RES = 6, A, B, QTE = 4,D

- 1 - متغيرات مرتبطة بالمتغيرات الشكلية - من حيث مواقعها في اللائحة . لدينا هكذا التناسب بين A و &U ، B و &V ، D و &X . إن فاصلتين متتاليتين تشيران إلى غياب متغير الوضع .
- 2 - متغيرات الكلمة المفتاح : الوصل بين المتغيرات الشكلية والفعلية القائم بفضل تشابه الاسم . هذه العناصر يجب أن يليها الرمز « = » ورّبّا قيمة تعدّل القيمة المحلّة نحو النقصان . في مثلنا تأخذ RES القيمة 6 ، QTE القيمة 4 ونحفظ NO بالقيمة 3 نحو النقصان .

3 - قد تكون لوائح متغيرات عمالة بأهّلة . لناخذ الماكرو تعليمة :

PROC 1	(A, B, C, D), K = (E, F, G, H)
--------	--------------------------------

والسطر النموذجي المناسب :

PROC 1	&APOS,&K=
--------	-----------

تتكوّن المتغيرات الفعلية بواسطة اللاحقين (A, B, C, D) و (E, F, G, H).  
 أما POS(3) فيستبدل عنقل بـ C خلال انتشار الماكرو تعليمية . كذلك يُستبدل  
 K(2) بـ F . بإمكان لوائح المتغيرات أن تكون ذات أطوال متغيرة ، وسنرى أن  
 الخاصية POS & N' تسمح بمعرفة طول اللائحة المرتبطة بـ POS .

### 2.2.22 . تطبيق

المثل التالي يقوم بتوليد تعليمات تسمح بجمع n خلية من الذاكرة منقولة إلى ماکرو  
 الإجراء بواسطة لائحة RES . ستحتوي عل النتيجة وNB تمثل عدد العناصر  
 المطلوب جمعها . المؤشر المركزي I يستعمل لمراجعة مختلف عناصر اللائحة .

```

1 MACRO
2 SONNE &MEN,&RES,&NB=&REG=
3 LCLA &I
4 L &REG,&MEN(1)
5 &I SETA
6 &BOUCLE ANOP
7 &I SETA &I+1
8 AIF (&I GT &NB).FIN
9 A &REG,&MEN(&I)
10 AGO &BOUCLE
11 ST &REG,&RES
12 PEND
  
```

```

000040 5030 C07A 00077A 64 SONNE (A,B,C,D),K,NB=4,REG=3
000044 5A30 C07B 00077B 65+ L 3,A
000048 5A30 C07C 00077C 66+ A 3,B
00004C 5A30 C089 000080 67+ A 3,C
000070 5030 C084 000084 68+ A 3,D
69+ ST 3,K
  
```

```

000074 72 A 05 P
000078 73 B 05 P
00007C 74 C 05 P
000080 75 D 05 P
000084 76 X 05 P
  
```

### 3.2.22 . الأمر MEXIT

يسمح بوقف تأويل الماكرو تعريف . من الممكن إختياره معادلاً للتفريع إلى الأمر

MEND .

### 4.2.22 . الأمر ACTR

يسمح بمراقبة عدد AIF وAGO الجاربي خلال التأويل المشروط . ويكتب :

ACTR (تعبير حسابي)

يؤدي إلى توليد عداد يعادل مضمونه قيمة للتعبير الحسابي . يمكن أن يكون العدد  
 مركزياً للماكرو تعريف أو شاملاً . في كل مرة يجري فيها تنفيذ AIF أو AGO بواسطة  
 المؤول ، فإن العداد المناسب لهذا القسم من البرنامج يُخفّض واحداً من  
 مضمونه . وعندما يبلغ الصفر ، فإن المؤول يخرج من الماكرو تعريف ( فعل معادل لـ

MEXIT) أو يُوقف التأويل إذا كان ذلك متعلقاً بعدد شامل . هذا الأمر يسمح بتحديد عدد الحلقات التي تجري في مرحلة ما قبل التأويل .

#### 5.2.22 . الأمر MNOTE

يمكن أن يُستعمل من قِبل المبرمج لتوليد رسالة الخطأ الخاصة به أو طباعة قيم وسيطة مأخوذة من متحولات التأويل .

ويمكن أن يكتب بعدة أشكال :

(1)	étiquette d'assemblage	MNOTE	code, 'message'
(2)	étiq. assem.	MNOTE	., 'message'
(3)	étiq. assem.	MNOTE	., 'message'
(4)	étiq. assem.	MNOTE	., 'message'

الكود هو عبارة عن تعبير حسابي بقيمة محصورة بين 0 و 255 يربط مستوى من الخطأ بالرسالة . في الشكل 2 يفترض بالكود أن يكون مُعادلاً لـ 1 . لا تُطبع الرسالة من ضمن رسائل الخطأ إلا إذا كان الكود الذي يشير إلى درجة الحقيقة هو أعلى من أو يعادل الكود المعتمد من المؤلف .

الشكلان 3 و 4 يولدان الرسالة كمجرد ملاحظة ..

#### 6.2.22 . الملاحظات :

من الممكن إنخال ملاحظات في ماكرو التعريفات على الشكل التالي :

\* COMMENTAIRE GERE

.. COMMENTAIRE NON GERE

#### 7.2.22 . الدوال من النوع الذاتي (Intrinsic)

##### &SYSLIST

تسمح ، داخل الماكرو تعريف ، بتسمية متغيرات الموقع الموجودة داخل ماكرو تعليمة النداء . وتكتب مؤشر أو مؤشرين يمكن أن يكونا عبارة عن تعابير حسابية من نوع ذلك الذي رأيناه في الفقرة 3.1.22 . سنختبر إستعمالها بالخاصية N .

&SYSLIST(&I) تشير إلى المتغير الفعلي الخاص بالموقع رقم i من التعليمة . يمكن أن يكون هذا المتغير الفعلي عبارة عن لائحة ( حسب الفقرة 3.1.22 ). في هذه الحالة ، سنسمي العنصر رقم j من اللائحة بالرتبة &I بواسطة &SYSLIST(&I,j) . في المثل المذكور في الفقرة 2.2.22(1,2) &SYSLIST(1,2) تعني المتغير B ، &SYSLIST(2) تعني X .

&SYSLIST(0) تعني الرسم الموجود قبل الماكرو تعليمة الخاصة بالنداء . هذه المهمة تسمح بتفادي تسمية للمتغيرات .

## &SYSNDX

هي عبارة عن عداد من أربعة أرقام عشرية ، وهو مركزي ضمن ماكرو - تعريف ، وتزداد قيمته عند كل استعمال جديد للماكرو . لا يمكن أن يُستعمل وحيداً ولكن يمكن أن يتحد مع رمز ما . هذه هي الوسيلة لتوليد أوسمة مختلفة عند كل نداء للماكرو . - التعريف وتسمح بتفادي الأخطاء في التأويل والنتيجة عن تعريف الرموز .  
مثلاً :

لفترض الماكرو - تعريف التالي :

```
MACRO
PROC SA,...
&A&SYSNDX
-----
-----
&B&SYSNDX
-----
-----
MEND
```

البناء الأول يتم بواسطة PROC ETIQ,...

المتحولة &A&SYSNDX تأخذ القيمة ETIQ0001

المتحولة &B&SYSNDX تأخذ القيمة R0001

في البناء الثاني بواسطة ... PROC ETIQ,...

المتحولة &A&SYSNDX تأخذ القيمة ETIQ0002

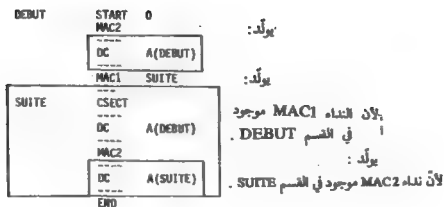
المتحولة &B&SYSNDX تأخذ القيمة R0002

## &SYSECT

تسمح بتعريف اسم القسم حيث توجد الماكرو - تعليمة للنادية . المثل التالي يوضح

ذلك :

```
MACRO
MAC1 &ETIQ
-----
&ETIQ CSECT
-----
DC A(&SYSECT)
-----
MAC2
-----
-----
MEND
MACRO
MAC2
-----
-----
DC A(&SYSECT)
-----
MEND
```



### &SYSPARM

يعطي وسيلة الرجوع إلى المتغير SYSPARM لبطاقة EXEC // في Job JCL (Control Language : لغة مراقبة العمل) .  
مثلاً :

```
// EXEC ASMC,PARM=SYSPARM(DEBUG)
//ASM.SYSIN DD *
TEST START 0
-----
AIF ('&SYSPARM' NE 'DEBUG'). (قراءة)
----- ولادة تعليمات
----- تنفيذ وتقرير
.SAUT ANOP
```

### &SYSTIME

يعطي ساعة التأويل بواسطة خمس سيات : h.h.mm

### &SYSDATE

يعطي التاريخ بواسطة ثمان سيات : mm/jj/aa

### 2.2.22 . الخصائص

مفهوم الخاصية المرتبط بمعنى أو بتعليمة جرت إثارته في الفقرة 2.3.6 . كما

إستعملنا الخاصية - طول ( فقرة 3.2.7 ) . يسمح المؤول لنا باستعمال خاصيات أخرى حيث البعض منها يجد إستعمالاً بسبب وجود إمكانيات التأويل للشروط .

#### الخاصية : TYPE T'

وقيمتها ممة أبجدية حسب نوع الرمز المُطبَّقة عليه . إذا كانت NUM ، مثلاً ، عبارة عن ثابتة عشرية موسعة ؛ فإن قيمة T'NUM ستكون Z . الحرف الذي يُميز النوع هو نفسه المُستعمل في الأوامر A:DC تناسب ثابتة عنوان من نوع A ، بينما B تناسب ثابتة منطقية ... ونضيف التناسبات التالية :

G	ثابتة بفاصلة ثابتة وطول محدد ظاهر
K	ثابتة بفاصلة متحركة وطول محدد ظاهر
R	ثابتة عنوان بطول محدد ظاهر
I	تعليمة - آلة
M	ماكرو تعليمة
CCW	W
J	اسم قسم
T	رمز خارجي
N	قيمة تعريف أوتوماتيكي
O	سمة مخلوطة

{ تتعلَّقان بمتغيرات الماكرو تعليمة

#### الخاصية L<sup>1</sup> LONGUEUR ( طول )

جرت دراستها في الفقرة 3.2.7 .

#### الخاصية مقياس S'

- عبارة عن قيمة رقمية تتعلَّق بنوع الرمز .
- لعدد عشري ( نوع P أو Z )
- عبارة عن عدد الأرقام في القسم الكسري .
- لعدد بفاصلة متحركة ( أنواع L, E, D أو K )

إنَّه عدد الأصفار السادس عشرية في يسار القسم العشري ( الوزن الأكبر ) .

- لعدد بفاصلة ثابتة ( الأنواع F ، M أو G )

عبارة عن القوة 2 التي يتم ضرب قيمة الثابتة بها . وتشير إلى عدد البتات في القسم الكسري إذا كان إيجابياً ، وعدد البتات المتروكة إذا كان سلبياً .

### الخاصية قسم صحيح P'

عبارة عن عدد يتعلّق بـ S' و L' .

$$P' = 2 * L' - S' - 1$$

- لعدد عشري من نوع P

$$P' = L' - S'$$

- لعدد عشري من نوع Z

$$L' \leq 8 \text{ مع } P' = 2 * (L' - 1) - S$$

- لعدد بفاصلة متحركة من نوع

$$K, L, E, D$$

$$L' > 8 \text{ مع } P' = 2 * (L' - 1) - S' - 2$$

- لعدد بفاصلة متحركة من نوع L

$$P' = 8 * L' - S' - 1$$

- لعدد بفاصلة ثابتة من نوع

$$G, F, H$$

### الخاصية عدد السيات K'

وتُطبّق فقط على متغيرات الماكرو - تعلّمة وأيضاً ، بإشراف OS ، على الرموز المتحركة ... & . وقُلّ ألدوال الذاتية (من نوع intrinsic) . وتعطي عدد سيات الرمز التي تطبّق عليه .

أمثلة : في مثل الفقرة 2.2.22 :  $K' \& MEM = 9$

$$\&A \text{ SETA } 253 : K' \& A = 3,$$

$$\&B \text{ SETB } 0 : K' \& B = 1,$$

$$\&C \text{ SETC 'ALPHA' : } K' \& C = 5.$$

### الخاصية عدد العناصر من اللائحة N'

وتطبّق فقط على متغيرات الماكرو - تعلّمة ، وتعطي عدد عناصر اللائحة .  
مثلاً :

PROC &A,&B,&K=  
PROC (1,2,,4),U,K=3

خط نموذج

ماكرو تعلّمة

N'&A = 4  
N'&SYSLIST = 2  
N'&SYSLIST(1) = 4.

(يتم تعداد السيات غير الموجودة)  
متغيرات الموقع

### 9.2.22 . أمثلة عن الماكرو - تعريفات

الماكرو - تعريف التالي يسمح بتوليد الأوامر (التوجيهات) المُعادلة لـ **RIEQUI**

12	MACRO	MACRO D-EQUIVALENCE REGISTERS	ماكرو معادل للرافف
13	EQUREG		
14	GRLA	AND	
15	SETA	1	
16	AND		
17	ST	(AND ST IS).FIN	
18	AND	EQU	AND
19	AND	SETA	AND+1
20	AND	AGO	ST
21	.FIN	MEMO	

## ويُؤد الكود التالي :

	33	EQUREG	ماكرو معادل المرافف
00001	34+4	MACRO D	
00002	35+R1	ESU 1	
00003	36+R2	ESU 2	
00004	37+R3	ESU 3	
00005	38+R4	ESU 4	
00006	39+R5	ESU 5	
00007	40+R6	ESU 6	
00008	41+R7	ESU 7	
00009	42+R8	ESU 8	
0000A	43+R9	ESU 9	
0000B	44+R10	ESU 10	
0000C	45+R11	ESU 11	
0000D	46+R12	ESU 12	
0000E	47+R13	ESU 13	
0000F	48+R14	ESU 14	
	49+R15	ESU 15	

الماكرو- تعريف PROLOGUE يسمح بشحن واحد أو عدة مرافف قاعدة مخزنًا مفهوم البرنامج المتادي حسب المعايير العادية للخدمة في الفصل 21 . وهو يُعرف في نفس الوقت منطقة SAVE AREA للبرنامج الجاري . عنوان القاعدة الذي جرى اختياره هو عنوان نقطة الدخول إلى البرنامج . ويرد الكود المؤد على الصفحة التالية .

```

14 MACRO
15 PROLOGUE &ARBASE=>&RBASE=
16 EQU 0
17 LCLLA &I,40
18 LCLC &CH
19 SYN 14,12,12(13)
20 إنشاء مصنع
21 &CH SETC 'ARBASE(13)'
22 &I SETA 2
23 &T1 AIF (&I GT N*ARBASE).SUIT1
24 &CH SETC '&CH*.*.*ARBASE(&I)'
25 &I SETA &I+1
26 AGO &T1
27 .SUIT1 ANOP
28 USING &ARBASE,&CH
29 LR &RBASE(13),15
30 ST 13,SAVEAREA+4
31 LR 2,13
32 LA 13,SAVEAREA
33 ST 13,8(2)
34 CONSTRUCTION DES CHARGEMENTS DES REGISTRES DE BASE
35 &I SETA 2
36 &T2 AIF (&I GT N*ARBASE).SUIT2 إنشاء شحن المرافف القاعدي
37 L 0,=F'4000'
38 &T3 AIF (&I GT N*ARBASE).SUIT2
39 AR 15,0
40 LR &RBASE(&I),15
41 &I SETA &I+1
42 AGO &T3
43 .SUIT2 ANOP
44 0 &+T6
45 SAVEAREA DS 10F
46 REND

```

من المفيد دراسة أمثلة الماكرو تعريفات المذكورة في كتاب [ـ. تابورييه Y.Tabourier ، أ. روشفلا Rochfeld ونس. فراك C. Frank ـ إنها عبارة عن ماكرو تعريفات تسمح ببناء برنامج مؤول بصورة بنوية مركبة . والكتاب يعرض للماكرو WHILE زائد شرط ، DO ، ENDWHILE ، IF ، THEN ، ELSE ، ENDIF ، BLOCK ، ENDBLOCK بالتفصيل ، وتقدم هي باستدعاء 2 ماكرو لتكرار مكسأ من المؤثرات .



## 23 . نصائح في البرمجة

ليس هدفنا عرض طريقة في البرمجة تشبه البرمجة الإنشائية ، ولكن ببساطة إعطاء بعض النصائح الناتجة عن الخبرة العملية لمختلف الطرق . هذه الملاحظات يمكن أن توسع لتشمل جميع أنواع المؤول وفي بعض الأحيان تنطبق عل اللغات المتطورة .

### 1.23 . تركيبة المعالجة

#### 1.1.23 . البرمجة الزجالية

هي عبارة عن قاعدة عامة في البرمجة . هناك فائدة من تقسيم المسألة إلى زجل ( أقسام ) صغيرة قدر الإمكان . كل زجلة تحمل مهمة معينة والبرنامج الرئيسي يؤمن ترابط الأقسام فيما بينها . ولقد عرضنا في الفصلين 20 و 21 . طريقة استعمال وسائل التقطيع وإنشاء البرامج - الثانوية .

#### 2.1.23 . تقليم وإعداد

البرنامج بلغة المؤول هو عادة عبارة عن نص غير واضح ، وبعد المصمم صعوبة في تعديل وإعادة قراءة ما كتبه منذ اللحظة التي يترك فيها برنامجاً جانباً لبعض الوقت . يجب إذا كتابة الملاحظات بعد كل تعليمة لتوضيح نص البرنامج . الأوامر SPACEN ( إدخال عدة أسطر » بيضاء ) ، EJECT ( عبور إلى الصفحة التالية ) و PRINT NOGEN ( إلغاء توليد كود الماكرو تعليمات ) تسمح بتسهيل نص البرنامج بجعله أكثر وضوحاً .

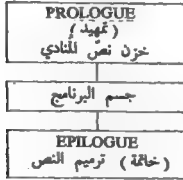
البرنامج المؤول بملاحظات يبدأ بتحديد مهمة الزجلة ، وروابطها مع الزجل الباقية كما يحتوي عل أسماء ومهمة المتحولات والمراصف المستعملة .

### 2.23 . تركيبة الزجلة

#### 1.2.23 . التمهيد والختام ( Prologue و epilogue )

بإمكاننا اعتبار كل برنامج وكأنه برنامج ثانوي لبرنامج آخر . الزجلة الرئيسية هي عبارة عن برنامج ثانوي من نظام التشغيل ويجب عليها أن تخزن نتائج البرنامج النهائي .

- تقنية الحزن وترميم نصّ المتادي هي أساسية وقد جرى تعريفها في الفصل 21 . بإمكاننا إنشاء كل زجلة حل الشكل التالي :



هناك فائدة للمبرمج في تحقيق التمهيدات والخاتمة الخاصة به حسب القواعد المتفق عليها والمذكورة في الفصل 21 . الاتصال بين الزجل المكتوبة في اللغات المختلفة سيكون مبسطاً وأكثر من خطأ سيتم تفاديه باستعمال مناسب للمراسف . لقد ذكرنا مثلاً في الفصل 22 الماكرو - تعليمة PROLOGUE التي تحلّ هذه المسألة وتوفّر حل المبرمج كتابة صعبة للتعليقات الأولية .

#### 2.2.23 . جسم البرنامج

يتألف من تعليمات قابلة للتنفيذ ومن معطيات . سنضع المعطيات بعد التعليقات . استعمال الأمر LTORG سيسمح لنا بوضع تأويل الثوابت من نوع حرفي في المكان الذي نرغب فيه . للتحويلات والثوابت ستكون إذا متراسّة ، مما يجعلها متجاورة في كل dump وستسمح بإجراء تقسيم سهل إلى أقسام إذا كنا نرغب بجعل البرنامج مبسطاً للتعديل والاختبار . سنستعمل عند الحاجة أوامر حجز مكان من الذاكرة بواسطة DC أكثر من بواسطة DS معطين بهذه الطريقة منطقة من الذاكرة بقيمة سوف يمكننا مراقبتها في dump (دلق) .

#### استعمال المرجعيات الرمزية

إن كتابة LR1,2 تعود عملياً إلى العمل بلغة الآلة . وفي المقابل فإن كتابة LR R1,R2 بعد تعزيف الرمز R1 و R2 بواسطة EQU معناها استعمال إمكانات ومرونة الترميز ، والمرجعان R1 و R2 يظهران في جدول الرموز . من الأفضل أيضاً إعطاء المراسف والتحويلات أسماء مكتوبة حرفياً كما جرى في أمثلة الفصل 35 . فليس من اللزج أكثر من قراءة التعليقات التي تذكر المراسف بشكل ظاهر .

هكذا ، فكتابة B \* + 14 تؤدي إلى سيئة تكمن في تجميد البرنامج ، ويصبح من

غير الممكن إدخال تعليمات جديدة بين العنواين \* 14+ و \* دون تعديل تعليمات الضريع . لذا فمن الأفضل تعريف وسم ALPHA وكتابة ALPHA B . الكتابات من النوع n + \* لا يجب أن تستعمل إلا داخل الماكرو - تعريفات . وختلماً يجب على البرنامج أن يكون دائماً مكتوباً مع أخذ التعديلات اللاحقة بعين الاعتبار إضافة إلى مسائل الصيانة ..

هكذا يجب تعريف جميع العناصر القابلة للتعديل في البرنامج بواسطة EQU . هذا الأمر هو شديد الأهمية . وفي حالة التعديل فهو يسمح بتخفيض عدد التغيرات المطلوب إجراؤها . ويقدم فائدة تكمن في جعل التعليمات « مزودة بملاحظات » . إن التمرين 8.13 يوضح لنا ذلك .

الخاصية - طول  
تسمح بجعل البرنامج يحتوي على متغيرات . كل تعديل على طول المنطقة لن يؤثر على التعليمات التي تذكر هذا الطول بواسطة L'ZONE .

تركيبة منطقة المعطيات  
بدلاً من مراجعة أقسام (field) نفس المنطقة بواسطة المسافة بالنسبة لبداية المنطقة ، من الأفضل تخصيص ( بواسطة EQU ) أسماء رمزية لمختلف هذه الأقسام . كل تعديل على التركيبة يصبح عندئذ سهلاً . يوضح لنا التمرين 2.8 تعريف تركيبة كهذه .

استعمال الكود الحرقي  
يترك للمؤول مهمة تعريف الثوابت الضرورية دون إسهاب . هذه الثوابت يمكن أن تكون مجموعة في المكان المطلوب بواسطة الأمر LTORG .

كتابة الأوسمة  
سنعرف الأوسمة بواسطة الأمر DSOH . نتأكد من تسطير (إصطفاف) حدود نصف - كلمة والوسم لن يعود مرتبطاً بالتعليمة الموجودة في الجهة المقابلة له . سيصبح ممكناً عكس بعض التعليمات بواسطة معالجة بسيطة للبطاقات .

استعمال المرافص  
قبل أية عملية برجة يجب التقبيل عن الخيارات التي يقوم بها النظام لاستعمال المرافص . وقد جرى عرض ذلك في الفصل 21 . وللمبرمج فائدة في إجراء نفس الاختيار لأسباب تتعلق بالتوافق . فلنذكر أن OS :

يشحن في R15 عنوان نقطة الدخول ،

في R14 عنوان العودة ،

في R13 عنوان المنطقة SAVE AREA .

و يستعمل R0 لإرسال نتيجة مهمة من نوع (FUNCTION في فورتران) ،

و R1 لإرسال عنوان لائحة متغيرات إلى برنامج ثانوي .

بعض التعليقات (TRT, EMDK) تستعمل المرصفين R1 و R2 . سيختار المبرمج مرادف القاعدة من ضمن المرادف 12 ، 11 و... و مرادف العمل من ضمن المرادف 3 ، 4 ، ...

إستعمال الماكرو - لغة (MACRO-language)

باستعمال الماكرو لغة فإن المؤلف يقترب من اللغة المتطورة . وهي تسمح للمبرمج بأن يكون مزوداً بوسائل إعداد البرنامج وجعله إنشائياً (مركباً) . وسيكون بإمكانه ، مثلاً ، إنشاء ماكرو - تعريف يسمح له بمتابعة أثر البرنامج عند التنفيذ بواسطة طباعة الأوسمة خلال مرحلة الاختبار . عند التأويل النهائي فإن توليد الماكرو - تعليمة سيتم إلغاؤه بواسطة تعديل بسيط لقيمة متحولة التأويل . ولن تولد أوسمة بواسطة ETIQ DS OH . بإمكان المبرمج أن يقوم أيضاً بإنشاء ماكروتعريفات تولد مثلاً : تعليمات من نوع WHILE ، DO ، ENDDO ... . وبإمكان الأوسمة أن تختفي من النص المطلوب تأويله ويصبح البرنامج أكثر إنشائياً .

وفي النهاية فإن الزجلة يمكن أن تحصل على التركيبة التالية :

MACRO-DEFINITIONS  
COMMENTAIRES  
EQU ...  
PROLOGUE  
CORPS  
EPILOGUE  
ZONE DE DONNÉES

ماكرو تعليمات  
ملاحظات  
EQU...  
مقدمة  
خاتمة  
جسم البرنامج  
منطقة المعطيات

3.23 . الخلاصة

بشكل عام لا نؤيد المبالغة في استعمال الحيل والحيلق من قبل المبرمج . فالبرنامج « التحايل » هو غامض على العموم بالنسبة للقارئ المبتدئ ، وأحياناً تقترب الحيل من الإضهار المبهم ويمكننا هنا تصور المشاكل التي قد تعترض عمل فريق صيانة البرامج .

في لغة المؤلف تختلف المسألة نوعاً ما . فبالإمكان إقامة عدد معين من الحيل ضمن نطاق تقنيات الحل وفي هذا الإطار يتعين على المبرمج أن يعرفها . لقد ذكرنا خلال الأمثلة والتأويل عدد كبيراً من الوصفات المنتشرة كغاية بشكل يسمح لنا باعتبارها كأدوات أساسية . هذا هو السبب الذي يجعلنا نصرّ على دراستها من قبل القارئ بعناية واهتمام .

## حلول التمارين

التميز 1.2 -	النظام 16	النظام 2	النظام 10
	F	1111	15
	100	10 0011	35
	400	1 0000 0000	256
	15C.8	100 0000 0000	1024
		1 0101 1100.1	348.5
التميز 2.2 -	النظام 2	النظام 10	النظام 16
	11 1010	50	3A
	1111 1111 1111	4095	FFF (-1000-1)
	1 1010 0011 1011	6715	1A3B
	1010 1011 1100	2748	ABC

التميز 3.2 - للكتمل إلى FFFF : ESC4  
للكتمل إلى ESC5:2

الطرح بواسطة جمع المكتمل إلى 2 (نتحقق ما إذا كان يحق لنا تجاهل المرحل) النتيجة : 1081 .

1A3B على 32 بتة : 00 00 1A 3B  
ESC5 على 32 بتة : FF FF E5 C5

- التميز 4.2 - تكويد الإشارة والقيمة المطلقة :  $4.16^7 + 16^6 + 15.16^5$  -  
 التكويد بالمكتمل إلى 2 :  $3.16^7 + 14.16^6 + 16^5$  -  
 التكويد بالفاصلة المتحركة :  $16^1 (15.16^{-1})$  -  
 المكس (الضد) بالإشارة والقيمة المطلقة : 41 F0 00 00  
 المكس بالمكتمل إلى 2 : 3E 10 00 00  
 المكس بالفاصلة المتحركة : 41 F0 00 00 (معايير)  
 لا يمكن لهذا التمثيل أن يكون تمثيل عدد مكود بالنظام DCB (عشري مكود ثنائياً) .

CS 03 20 00 =  $-16^5(3.16^{-2}+2.16^{-3})$  - تمرين 5.2  
 =  $- \frac{16}{16} .16(3.16^{-2}+2.16^{-3})$   
 =  $- 16^4(3.16^{-1}+2.16^{-2}) = 04 32 00 00$

TAB DC 100AL1(=-TAB+1) - تمرين 1.8  
 TAB DC 100A((-TAB)/4+1)

NOSS DS OCL13 L'NOSS = 13  
 SEXE DS CL1 L'SEXE = 1  
 DATE DS OCL4 L'DATE = 4  
 ANNEE DS CL2 L'ANNEE = 2  
 MDIS DS CL2 L'MDIS = 2  
 LIEUNAI DS OCL5 L'LIEUNAI = 5  
 DEPART DS CL2 L'DEPART = 2  
 COM DS CL3 L'COM = 3  
 NO DS CL3 L'NO = 3

Z1 DS OF تأخير على حد كلمة  
 PRIX DS OCL12 ZL8  
 QTE DS ZL4  
 ZZ ORG Z1  
 NO DS OCL14  
 TEXTE DS F  
 CL10

- تمرين 1.9

L'IC	OBJECT CODE	ADDR1	ADDR2	SYMT	SOURCE STATEMENT
000000				1	CSECT
000000	S810 C22C	0002C	00000	2	USING *12
000000	S833 C02C	0002C		3	L 8*0
000000	0000			4	L 3*0(3)
000000	*** ERROR ***			5	L 8*0
000000	0000 0300	00004		6	ST 0*8*4*(3*0)
000000	*** ERROR ***			7	L 8*8*1011*(3)
000000	S803 0000	00008		8	L 0*E(8)
000000	*** ERROR ***			9	L A*E(8)
000000	S801 C040	00040		10	MVC A(8*0),0
000000	S801A D200	00000	0002C	11	E(1*0),0
000000	S803 C040	00040	0002C	12	L 2*0+L*0
000000	S820 C030	00030		13	EOU 0
000000	00000			14	EOU 1
000000	0000A			15	EOU 10
000000				16	OS 5F
000000				17	OS 12F
000000				18	END

# ASSEMBLER DIAGNOSTICS AND STATISTICS

SYMT	ERROR CODE	MESSAGE
8	IF0217	RELOCATABILITY ERROR NEAR OPERAND COLUMN 4
8	IF0217	RELOCATABILITY ERROR NEAR OPERAND COLUMN 2
8	IF0217	RELOCATABILITY ERROR NEAR OPERAND COLUMN 2

NUMBER OF STATEMENTS FLAGGED IN THIS ASSEMBLY = 3  
 HIGHEST SEVERITY WAS 12

تمرين 1.11 -

حلول أخرى بواسطة «أو المختصرة» أو «الإزاحات» .  

$$\begin{array}{cc} LA & R,0 \\ SR & R,R \end{array}$$

تمرين 2.11 -

$LCR \quad R,R$

تمرين 3.11 -

نستعمل كون نصف الكلمة موسعاً إلى كلمة قبل العملية بواسطة انتشار  
 بث ذات وزن قوي .  

$$LR \quad R, -N^2 - 1^2$$

تمرين 4.11 -

$$\begin{array}{cc} LA & R,2048 \\ LA & R,4095 \\ L & R,-F'4096 \end{array} \quad \text{أو:} \quad \begin{array}{c} (LA \ R,4095 \\ (LA \ R,1(0,R) \end{array}$$

تمرين 5.11 -

$LA \quad R,4(0,R)$

تمرين 6.11 -

$$\begin{array}{cc} MV1 & ZONE,C'a' \\ MVC & ZONE+1(L'ZONE-1),ZONE \end{array}$$
  
 نستعمل كون الحركة تتم بلمة بعد بلمة من اليسار إلى اليمين .

تمرين 1.12 -

$$\begin{array}{ccc} N & DC & \dots \\ R1 & EQU & 3 \\ & --- & \\ & L & R1,N \\ TRAIT & --- & \\ & --- & \\ BCT & R1,TRAIT & \end{array}$$
  
 عدد التكرارات .  
 معالجة التكرار .

تمرين 2.12 -

تسمح الماكرو تعليمة SNAP بالحصول على عمليات دلق («dumps») جزئية في الذاكرة . ويجب أن تسبقها ماكرو OPEN (فتح سجل) . في حالتنا الحاضرة يمتد الدلق dump من العنوان SNAPDEB حتى العنوان SNAPFIN . تعطي الكلمة PSW عنوان بداية SNAP . وتعطي الجهة

البحق من dump ، حتى يكون ذلك ممكناً ، تفسير محتوى الذاكرة الثاني  
 على شكل سجلات . وسيتمرن القارئ بمحاولة إيجاد محتويات مختلف مناطق  
 البرنامج عبر حساب العناوين من خلال العنوان الأساسي الموجود في  
 الموصف 12 .

( انظر اللاحقة listing في الصفحة اللاحقة ) .

تمرين 3.12-

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
000000				1	DEBUT	START 0
				2		PRINT NOGEN.DAT
				3		عكس سلسلة من السجلات *
			000003	4	WORK	EQV 3
			000004	5	IND1	EDU 4
			000005	6	INC2	EDU 5
				7	SHAPDES	OS ON
				8	PROLOGUF	OS ON
000000				9		SYN (4-12-12(13)
000000	90EC 008C	0C00C	00000C	10	USING	DEBUT.12
000004	10CF			11	LS	12-15
000006	5000 C078	0C078		12	SY	13-SAVE+4
00000A	4100 C074	0C074		13	LA	13-SAVE
00000E	4150 C000	000000		15	LA	INC2,0
000012	4140 0005	000005		16	LA	INC1,L*CH1
000016				17	BCL	OS ON
000016	4334 C067	000067		18	IC	WORK.CH1-1(IND1)
00001A	4235 CC60	000060		19	STC	WORK.CH2(IND2)
00001E	4155 00C1	0000C1		20	LA	INC2,1(IND2)
000022	4640 C016	000016		21	RCT	INC1,BCL

تمرين 4.12- BCTR R,0

تمرين 5.12- BALR R,0

تمرين 1.13- XC ZONE,ZONE منطقة بطول L  
 XR R1,R1 المرصف  
 RI OCTET,X'00' بايت

تمرين 2.13- لفترض التعليمة في العنوان INSTR . إذن يوجد كود الطول في INSTR  
 + 1 ( تعليمة بنسق SS ) .

RI (إعادة تصفير)

RI (حيث XX هو الطول ناقص واحد)

علينا أن نتذكر أنه ، بالنسبة للتعليقات من النوع SS ، الطول المؤول هو  
 الطول ناقص واحد .

تمرين 3.13-

XC ZONE1(L),ZONE2  
 XC ZONE2(L),ZONE1  
 XC ZONE1(L),ZONE2

XR R1,R2  
 XR R2,R1  
 XR R1,R2

PACK OCTET,OCTET أو UNPK OCTET,OCTET



تحويل 4.13 - كما في التحويل 6.11 ، نستعمل كون العمليات مع التعليلات MVC ،  
CLC ... نحري باية بعد باية مع انتشار من اليسار إلى اليمين .

CL.I ZONE, X'DD' (مقارنة أول باءة مع 0)  
BNE NONZERO  
CL.C ZONE+1(L'ZONE-1),ZONE  
BNE NONZERO  
B.F ZERO

التعرف إلى الفراغ (blank) يتم عبر المقارنة مع X40.

### تحریریں 5.13۔

X1      ←5,X'FO'  
NOP      ET10

### - 6.13: تمرین

ETIQ  
#3.X'ED'

تمرين 7.13 . INCRE وREF يشكّان مرصفاً مزدوجاً يحتوي الزيادة والرجع بالنسبة للتعليمية BXL وNOMBRE هو عنوان العدد. أما PTR فهو مرصف مصوّب (مؤسّر) .

```

                                NOMBRE DE
                                +
                                FD  FD  FD  FD  F2  F1
                                +PTR                                +REF
LA      PTR,NOMBRE
LA      REF,L'NOMBRE-1(PTR)
LA      INCR,1
TEST   CL1 0(PTR),C'0'
ONE     SUITE
WVI     0(PTR),C' '
SNITE   PTR,INCR,TEST

```

**8.13**

INDIC	DC	X'00'
INXLEC	EQU	X'80'
INDECR	EQU	X'40'
INDMAT	EQU	X'20'
01	INDIC, INDMAT	
01	INDIC, INXLEC+INDECR	
HI	INDIC, X'FF'-(INXLEC+INDECR)	
TH	INDIC, INDMAT	
D0	ALPHA	
TH	INDIC, INXLEC+INDMAT	
D0	BETA	
HI	GAMMA	
DZ	DELTA	

مع هذا الحل فإن التعديل المتعلق بـ INDLEC يُترجم بواسطة :

INDLEC EQU X'01'

لا تتأثر أي تعليمة لتحديد موضوع أو اختيار . والأمور لا يكون كذلك إن نحن لم نستعمل EQU لتحديد المؤشرات الثنائية ، فحينئذٍ لكان الكود مجسداً بسبب ظهور القيم X'80' ... في قلب التعليمات نفسها .  
من جهة أخرى فإن هذه التقنية تحول التعليمات لأن تصبح مؤلفة ذاتياً .

تمرين 1.14 -

SLL R,32  
SRL R,32 أو

تمرين 2.14 -

SLA R,3 : 2<sup>3</sup> يكتب الضرب بـ  
SRA R,4 : 16 القسمة على

ترافق القسمة عملية بتر (قطع) . والتدوير بالمكمل إلى 2 يعادل  $15/2 +$   
تعطي 7 في حين أن  $15/2 -$  تعطي 8 - .

تمرين 3.14 - SLDA R,0  
ZERO  
هو مصرف مزدوج

تمرين 4.14 . أثناء عملية إزاحة دائرية إلى اليسار نحاول إعادة إدخال كلّ بتة خارجة في جهة اليمين . العمل يتم على مصرف مزدوج . بعد تصغير مصرف اليسار نجرى إزاحة مزدوجة بشكل يسمح بأن نجد من جديد في مصرف اليسار البتات المفقودة في مصرف اليمين . ونتيج لنا تعليمة أو (OR) بإعادة وضعها في مصرف اليمين . هنا نجرى إزاحة دائرية من أربعة مواقع على المصرف 7 .

SLL 6,32 تصغير  
SLL 6,4  
OR 7,6

تمرين 1.18 -

نستعمل تعليمة TR « بالقلب »

TR	CLE,ARTICLE	
ARTICLE	BC	CL10'ABCDEFGHIJ' (لقرة)
CLE	DS	OCL5 (مفتاح)
	DC	HL1'S,B,7,1,2

## تمارين 2.18 -

نستخدم المعلومات

TR CHAINE (B), TABLE

---

DC C'0123456789ABCDEF'

( جدول )

DC 2F

( سلسلة )

## ملحقات

جدول تكويد السهات  
جدول أبجدي للعمليات  
أوامر المؤول  
مميزات الثوابت  
كود حرفي (تذكيري) موضع

جدول تكرید السیات

بطاقة مفتوحة	سمة مطبوعة	حرفي تكريري	سلسل عشري	بطاقة مفتوحة	سمة مطبوعة	حرفي تكريري	سلسل عشري
0	00		12-0-0-1	64	40	STH	لا تقيس
1	01		12-0-1	65	41	LA	12-0-0-1
2	02		12-0-2	66	42	STC	12-0-0-2
3	03		12-0-3	67	43	IC	12-0-0-3
4	04		12-0-4	68	44	EX	12-0-0-4
5	05	SPM	12-0-5	69	45	NAL	12-0-0-5
6	06	BALR	12-0-6	70	46	BCT	12-0-0-6
7	07	BCTR	12-0-7	71	47	BC	12-0-0-7
8	08	SSK	12-0-8	72	48	LH	12-0-0-8
9	09	ISK	12-0-9	73	49	CH	12-1-1
10	0A	SVC	12-0-0-2	74	4A	AH	12-0-2
11	0B		12-0-3	75	4B	SH	12-0-3
12	0C		12-0-4	76	4C	MH	12-0-4
13	0D		12-0-5	77	4D		12-0-5
14	0E	MYCL	12-0-6	78	4E	CYD	12-0-6
15	0F	CLCL	12-0-7	79	4F	CVB	12-0-7
16	10	LPR	12-11-0-1	80	50	ST	12
17	11	LNR	11-0-1	81	51		12-11-0-1
18	12	LTR	11-0-2	82	52		12-11-0-2
19	13	LCA	11-0-3	83	53		12-11-0-3
20	14	NL	11-0-4	84	54	N	12-11-0-4
21	15	CLA	11-0-5	85	55	CL	12-11-0-5
22	16	OR	11-0-6	86	56	O	12-11-0-6
23	17	XR	11-0-7	87	57	X	12-11-0-7
24	18	LR	11-0-8	88	58	L	12-11-0-8
25	19	CR	11-0-0-1	89	59	C	11-0-1
26	1A	AR	11-0-0-2	90	5A	A	11-0-2
27	1B	SR	11-0-0-3	91	5B	S	11-0-3
28	1C	MR	11-0-0-4	92	5C	M	11-0-4
29	1D	DR	11-0-0-5	93	5D	D	11-0-5
30	1E	ALR	11-0-0-6	94	5E	AL	11-0-6
31	1F	SLR	11-0-0-7	95	5F	SL	11-0-7
32	20	LPDR	11-0-0-0-1	96	60	STD	11
33	21	LNDR	0-0-1	97	61		0-1
34	22	LTDR	0-0-2	98	62		11-0-2
35	23	LCDR	0-0-3	99	63		11-0-3
36	24	HDR	0-0-4	100	64		11-0-4
37	25	LADR	0-0-5	101	65		11-0-5
38	26	MXR	0-0-6	102	66		11-0-6
39	27	MXDR	0-0-7	103	67	MXD	11-0-7
40	28	LDR	0-0-8	104	68	LD	11-0-8
41	29	CDR	0-0-0-1	105	69	CD	0-0-1
42	2A	ADR	0-0-0-2	106	6A	AD	12-11
43	2B	SDR	0-0-0-3	107	6B	SD	0-0-3
44	2C	MDR	0-0-0-4	108	6C	MD	0-0-4
45	2D	DDR	0-0-0-5	109	6D	DD	0-0-5
46	2E	AWR	0-0-0-6	110	6E	AW	0-0-6
47	2F	SWR	0-0-0-7	111	6F	SW	0-0-7
48	30	LNER	12-11-0-0-0-1	112	70	STE	12-11-0
49	31		9-1	113	71		12-11-0-0-1
50	32	LYER	9-2	114	72		12-11-0-0-2
51	33	LCRR	9-3	115	73		12-11-0-0-3
52	34	HRR	9-4	116	74		12-11-0-0-4
53	35	LRER	9-5	117	75		12-11-0-0-5
54	36	AXR	9-6	118	76		12-11-0-0-6
55	37	EXR	9-7	119	77		12-11-0-0-7
56	38	LER	9-8	120	78	LE	12-11-0-0-8
57	39	CHR	9-0-1	121	79	CE	8-1
58	3A	ARR	9-0-2	122	7A	AE	8-2
59	3B	SHR	9-0-3	123	7B	SE	8-3
60	3C	MER	9-0-4	124	7C	ME	8-4
61	3D	DER	9-0-5	125	7D	DE	8-5
62	3E	AUR	9-0-6	126	7E	AU	8-6
63	3F	SUR	9-0-7	127	7F	SU	8-7

جدول تكويد السيات

بطاقة مطبوعة	يا سمة مطبوعة	حرفي تذكيري	سلسل عشري	19 ن	بطاقة	سمة مطبوعة	حرفي تذكيري	سلسل عشري	عشري
128			SSM	192	CO			128	
129				193	CI			129	
130			LPSW	194	C2			130	
131				195	C3			131	
132			WRD	196	C4			132	
133			RDD	197	C5			133	
134			BXK	198	C6			134	
135			BXLE	199	C7			135	
136			SRL	200	C8			136	
137			SL	201	C9			137	
138			SRA	202	CA			138	
139			SLA	203	CB			139	
140			SRDL	204	CC			140	
141			SLDL	205	CD			141	
142			SRDA	206	CE			142	
143			SLDA	207	CF			143	
144			STM	208	D0			144	
145			TM	209	D1	MVN	J	145	
146			MYI	210	D2	MVC	K	146	
147			TS	211	D3	MVZ	L	147	
148			NI	212	D4	NC	M	148	
149			CLJ	213	D5	CLC	N	149	
150			OI	214	D6	OC	O	150	
151			XI	215	D7	XC	P	151	
152			LM	216	D8		Q	152	
153				217	D9		R	153	
154				218	DA			154	
155				219	DB			155	
156			SHO	220	DC	TR		156	
157			TIO	221	DD	TKT		157	
158			HIO	222	DE	ED		158	
159			TCN	223	DF	EDMK		159	
160				224	ED			160	
161				225	E1			161	
162				226	E2			162	
163				227	E3			163	
164				228	E4			164	
165				229	E5			165	
166				230	E6			166	
167				231	E7			167	
168				232	E8			168	
169				233	E9			169	
170				234	EA			170	
171				235	EB			171	
172				236	EC			172	
173			STNSM	237	ED			173	
174			STOSM	238	EE			174	
175			SIOP	239	EF			175	
176			MC	240	EG			176	
177				241	EH			177	
178			LBA	242	FI	SRP		178	
179				243	F2	MVO		179	
180				244	F3	PACK		180	
181				245	F4	UNPK		181	
182				246	F5			182	
183			STCTL	247	F6			183	
184			LCCTL	248	F7			184	
185				249	F8			185	
186				250	F9			186	
187			CS	251	FA			187	
188			CD8	252	FB			188	
189				253	FC			189	
190			CLM	254	FD			190	
191			STCM	255	FE			191	
			ICM		FF				

# جدول أبجدي للتعليمات

النسق	منطقة العوامل	Format	منطقة العوامل
RR	$R_1, R_2$	SI	$D_1(B_1), I_2$
RR-M	$M_1, R_2$		
RR-1	$R_1$	S	$D_2(B_2)$
RR-1	$I_1$		
RX	$R_1, D_2(X_2, B_2)$	SS-1	$D_1(L, B_1), D_2(B_2)$
RX-M	$M_1, D_2(X_2, B_2)$	SS-2	$D_1(L_1, B_1), D_2(L_2, B_2)$
		SS-3	$D_1(L_1, B_1), D_2(B_2), I_3$
RS	$R_1, R_3, D_2(B_2)$	مواصف X, R, إزاحة D مفتاح بأربعة بتات M قيمة ثنائية I طول L	
RS-M	$R_1, M_3, D_2(B_2)$		

الذالة (الوظيفة)	حرفي تدكييري	CCP سلس عشري	النسق	موضع CC
Add	AR	1A	RR	*
Add	A	5A	RX	*
Add Decimal	AP	FA	SS-2	*
Add Halfword	AH	4A	RX	*
Add Logical	ALR	1E	RR	*
Add Logical	AL	5E	RX	*
AND	NR	1A	RR	*
AND	N	5A	RX	*
AND	NI	9A	SI	*
AND	NC	D4	SS-1	*
Branch and Link	BALR	05	RR	
Branch and Link	BAL	45	RX	
Branch on Condition	BCR	07	RR-M	
Branch on Condition	BC	47	RX-M	
Branch on Count	BCTR	06	RR	
Branch on Count	BCT	46	RX	
Branch on Index High	BXH	06	RS	
Branch on Index Low or Equal	BXLE	87	RS	
Compare	CR	19	RR	*
Compare	C	59	RX	*
Compare and Swap	CS	BA	RS	*
Compare Decimal	CP	F9	SS-2	*
Compare Double and Swap	CDS	0B	RS	*
Compare Halfword	CH	49	RX	*
Compare Logical	CLR	15	RR	*
Compare Logical	CL	55	RX	*
Compare Logical	CLC	D5	SS-1	*
Compare Logical	CLJ	95	SI	*
Compare Logical Characters under Mask	CLM	BD	RS-M	*
Compare Logical Long	CLCL	0F	RR	*
Convert to Binary	CVB	4F	RX	
Convert to Decimal	CVD	4E	RX	

البيانات ( الوطنية ) Fonction	حرفي تذكيري Alphabétique	ساعات عشرية hexa- decimal	النسق Format	محدد موضع CC
Divide	DR	1E	RR	
Divide	D	5D	RX	
Divide Decimal	DP	FD	SS-2	
Edit	ED	DE	SS-1	*
Edit and Mask	EDMK	DF	SS-1	*
Exclusive OR	XR	17	RR	*
Exclusive OR	X	57	RX	*
Exclusive OR	XI	97	SI	*
Exclusive OR	XC	D7	SS-1	*
Execute	EX	44	RX	
Insert Character	IC	43	RX	
Insert Characters under Mask	ICM	8F	RS-M	*
Load	LR	18	RR	
Load	L	58	RX	
Load Address	LA	41	RX	
Load and Test	LTR	12	RR	*
Load Complement	LCR	13	RR	*
Load Halfword	LH	48	RX	
Load Multiple	LH	98	RS	
Load Negative	LNP	11	RR	*
Load Positive	LPR	10	RR	*
Monitor Call	MC	AF	SI	
Move	MV	92	SI	
Move	MVC	D2	SS-1	
Move Long	MVCL	0E	RR	*
Move Numerics	MVN	D1	SS-1	
Move with Offset	MVO	F1	SS-2	
Move Zones	MVZ	D3	SS-1	
Multiply	MR	1C	RR	
Multiply	M	5C	RX	
Multiply Decimal	MP	FC	SS-2	
Multiply Halfword	MH	4C	RX	
OR	OR	16	RR	*
OR	O	56	RX	*
OR	OI	96	SI	*
OR	OC	D6	SS-1	*
Pack	PACK	F2	SS-2	
Set Program Mask	SPM	04	RR-1	
Shift and Round Decimal	SRP	F0	SS-3	*
Shift Left Double	SLDA	8F	RS	*
Shift Left Double Logical	SLDL	8D	RS	
Shift Left Single	SLA	88	RS	*
Shift Left Single Logical	SLL	89	RS	
Shift Right Double	SRDA	8E	RS	*
Shift Right Double Logical	SRDL	8C	RS	
Shift Right Single	SRA	8A	RS	*
Shift Right Single Logical	SRL	88	RS	
Store	ST	50	RX	
Store Character	STC	42	RX	
Store Characters under Mask	STCM	8E	RS-M	
Store Clock	STCK	8205	S	*
Store Halfword	STH	40	RX	
Store Multiple	STM	90	RS	

Subtract	SR	1B	RR	*
Subtract	S	5B	RX	*
Subtract Decimal	SP	FB	SS-2	*
Subtract Halfword	SH	4B	RX	*
Subtract Logical	SLR	1F	RR	*
Subtract Logical	SL	5F	RX	*
Supervisor Call	SVC	0A	RR-1	*
Test and Set	TS	93	S	*
Test under Mask	TM	91	SI	*
Translate	TR	DC	SS-1	*
Translate and Test	TRT	DD	SS-1	*
Unpack	UNPK	F3	SS-2	*
Zero and Add Decimal	ZAP	FB	SS-2	*

### تعليمات حسابية بالفاصلة للحركة

Add Normalized, Extended	AXR	36	RR	*
Add Normalized, Long	ADR	2A	RR	*
Add Normalized, Long	AD	6A	RX	*
Add Normalized, Short	AER	3A	RR	*
Add Normalized, Short	AE	7A	RX	*
Add Unnormalized, Long	AWR	2E	RR	*
Add Unnormalized, Long	AW	6E	RX	*
Add Unnormalized, Short	AUR	3E	RR	*
Add Unnormalized, Short	AU	7E	RX	*
Compare, Long	CDR	29	RR	*
Compare, Long	CD	69	RX	*
Compare, Short	CER	39	RR	*
Compare, Short	CE	79	RX	*
Divide, Long	DDR	2D	RR	*
Divide, Long	DD	6D	RX	*
Divide, Short	DER	3D	RR	*
Divide, Short	DE	7D	RX	*
Halve, Long	HDR	24	RR	*
Halve, Short	HE	34	RR	*
Load and Test, Long	LTDR	22	RR	*
Load and Test, Short	LTER	32	RR	*
Load Complement, Long	LCDR	23	RR	*
Load Complement, Short	LCER	33	RR	*
Load, Long	LDR	28	RR	*
Load, Long	LD	68	RX	*
Load Negative, Long	LNDR	21	RR	*
Load Negative, Short	LNER	31	RR	*
Load Positive, Long	LPDR	20	RR	*
Load Positive, Short	LPER	30	RR	*
Load Rounded, Extended Long	LRDR	25	RR	*
Load Rounded, Long to Short	LRER	35	RR	*
Load, Short	LER	38	RR	*
Load, Short	LE	78	RX	*
Multiply, Extended	MXR	26	RR	*
Multiply, Long	MDR	2C	RR	*
Multiply, Long	MD	6C	RX	*
Multiply, Long/Extended	MXDR	27	RR	*
Multiply, Long/Extended	MXD	67	RX	*
Multiply, Short	MER	3C	RR	*
Multiply, Short	ME	7C	RX	*
Store, Long	STD	60	RX	*

Store, Short	STE	70	RX	
Subtract Normalized, Extended	SXR	37	RR	*
Subtract Normalized, Long	SDR	2B	RR	*
Subtract Normalized, Long	SD	6B	RX	*
Subtract Normalized, Short	SER	3B	RR	*
Subtract Normalized, Short	SE	7B	RX	*
Subtract Unnormalized, Long	SMR	2F	RR	*
Subtract Unnormalized, Long	SW	6F	RX	*
Subtract Unnormalized, Short	SUR	3F	RR	*
Subtract Unnormalized, Short	SU	7F	RX	*

### لوامر المؤول

تعريف المعطيات	DC DS COM	Assemblage conditionnel	تعريف مشروط	MACRO MMOTE MEXIT MEMD
تقطيع	START CSECT DSECT COM ENTRY EXTN			ACTR AGO AIF AWOP GBLA GBLB GBLC LCLA LCLB LCLC SETA SETB SETC
تعريف المراجع القاعدية	USING DROP			
مراقبة اللاحقة	TITLE EJECT SPACE PRINT			
مراقبة البرنامج	END ORG LTORG CNOB END COPY PUNCH REPRO ISEQ ICTL PUSH POP OPSYN			

# مميزات الثوابت

التنوع	الطول الضميني	حد الإصطقات	يتميز بـ	يرتبط إلى
C	-	بلاية	سبات	اليمن
X	-	بلاية	أرقام سانس عشرية	اليسار
B	-	بلاية	أرقام ثنائية	اليسار
F	4	كلمة	أرقام عشرية	اليسار
H	2	نصف كلمة	أرقام عشرية	اليسار
E	4	كلمة	أرقام عشرية	اليمن
D	8	كلمة مزدوجة	أرقام عشرية	اليمن
L	16	كلمة مزدوجة	أرقام عشرية	اليمن
P	-	بلاية	أرقام عشرية	اليسار
Z	-	بلاية	أرقام عشرية	اليسار
A	4	كلمة	تعبير	اليسار
V	2	نصف كلمة	تعبير	اليسار
S	2	نصف كلمة	تعبير	-
V	4	كلمة	مز قابل للتخل	اليسار

## الكود الحرفي موسع

القطاع	التعليمة للولادة	للحقف ...	كود العملية الحرفي
1111	BC 15.... BCR 15....	تفرع غير مشروط	B ... BR ...
0000	BC 0.... BCR 0....	لا عملية	NOP ... NOPR ...
... بعد تعليمات المقارنة			
0010	BC 2.... BCR 2....	تفرع إذا كان : (*) 2 المقتر > المقتر 1 (*) 0p	BN ... BNR ...
0100	BC 4.... BCR 4....	" < "	BL ... BLR ...
1000	BC 8.... BCR 8....	" = "	BE ... BER ...
1101	BC 13.... BCR 13....	" ≤ "	BNH ... BNHR ...
1011	BC 11.... BCR 11....	" ≥ "	BNL ... BNLR ...
0111	BC 7.... BCR 7....	" ≠ "	BNE ... BNER ...
... بعد التعليمات الحسابية			
0001	BC 1.... BCR 1....	تفرع إذا كانت النتيجة يفض عن السعة ...	BO ... BOR ...
0010	BC 2.... BCR 2....	... > 0	BP ... BPR ...
0100	BC 4.... BCR 4....	... < 0	BN ... BNR ...
1101	BC 13.... BCR 13....	... ≤ 0	BNP ... BNPR ...
1011	BC 11.... BCR 11....	... ≥ 0	BNH ... BNHR ...
0111	BC 7.... BCR 7....	... = 0	BNZ ... BNZR ...
1000	BC 8.... BCR 8....	... = 0	BZ ... BZR ...

(\*) المقصود هنا المقتران 1 و 2 في تعليمة المقارنة .

ملاحظة : الكود الحرفي التذكيري المنتهي بحرف ٧ ، يؤد تعليمات من النسق RR . للرصف المذكور يحتوي على عنوان التفرع .

مثلا : BR 3 : تفرع غير مشروط إلى العنوان الواقع في الرصف ٣ .

B ALPHA : تفرع غير مشروط إلى العنوان ALPHA .



## ترجمة الملاحظات الواردة في بعض البرامج الموجودة في الكتاب

السطر	الملاحظة	الصفحة
5	توابت مهات . لا يوجد اصطفاك خامس . الطول 256	69
6	تأطير إلى اليسار . يتر إلى اليمين	
8	يتر إلى اليمين .	
9	تأطير إلى اليسار تكمله فراخفت .	
11	توليد فاصلة عليا واحدة .	
11	نفس الملاحظة	
12	تكرار ووتر	
15	توابت سافس عشرية . تأطير إلى اليمين . يتر إلى اليسار .	
16	طول ضمني	
17	طول ظاهر .	
18	يتر .	
21	توابت ثنائية . الطول الأقصى 256 بايت تأطير إلى اليمين .	
22	تكمله اصفاك إلى اليسار . اصطفاك على البايته .	
23	ثنائي	
24	نتر إلى اليسار .	
25	نتر .	
26	تكرار .	
29	توابت بالفاصلة الثمانية على كلمة (F) أو نصف كلمة (H) .	
30	اصطفاك على الكلمة أو نصف الكلمة . عندما يكون الطول	
31	محددا لا يعود هناك اصطفاك . الثمانية هي بالنظام العشري	
34	إزاحة 3 بتات إلى اليسار ( 8 )	
36	إزاحة 3 بتات إلى اليمين (/8)	
39	مدور أعلى	
40	مدور أصغر .	
42	تعديل الطول LONG والاصطفاك ALIGN .	
43	إزاحة بتين إلى اليسار .	
49	توابت بالفاصلة للحركة والبنقة البسيطة . اصطفاك على الكلمة	70
50	تأطير إلى اليمين . لا يتر . القيمة مدقورة .	
51	الطول الضمني 4 بايتات .	

- 52 بنافذة متحركة  
57 ثوابت بالفاصلة المتحركة وبالذقة المزدوجة  
58 اصطفاك على الكلمة المزدوجة . تأطير إلى اليمين . لا يتر  
59 اللقمة مدوّرة . الطول الضمني 8 بايتات  
66 ثوابت بالفاصلة المتحركة وبالذقة الرباعية  
67 اصطفاك على الكلمة المزدوجة . الطول الضمني 16 بايتة .  
68 لا يتر . اللقمة مدوّرة . -أس من 85- إلى 75+ .  
73 ثوابت عشوية . الطول الأقصى يبلغ  
74 16 بايتة . الإشارة تقع في الربع الأيسر  
75 من البايتة اليمنى الأخيرة . تأطير إلى اليمين . يتر إلى اليسار .  
76  $X^*F$  أو  $X^*C$  في موقع الإشارة يعتبران مثل +  
77  $X^*E$  أو  $X^*D$  في موقع الإشارة يعتبران مثل -  
78 لا يتم ترجمة الفاصلة المشربة أبداً إلى التثني .  
79 تأطير إلى اليمين . يتر إلى اليسار .  
80 الطول الضمني .  
82 يتر إلى اليسار  
86 ثوابت المشربة المكثفة (Packed)  
87 نفس قواعد الثوابت السابقة .  
88 تقع الإشارة في الربع الأيمن الأخير .

- 2 رمز خارجي  
6 رمز قابل للتقل  
9 ثوابت عنوان من النوع A  
10 تُكتب DC A (تعبير مطلق أو قابل للتقل)  
11 اصطفاك على الكلمة . الطول الضمني 4 بايتات .  
12 الأطوال الظاهرة للمكثفة هي من 1 إلى 4 بايتات .  
13 يتر إلى اليسار . يمكن التحديد في كود حرقي .  
18 طول ظاهر  
20 رمز محوسبي  
23 ثوابت عنوان من النوع Y  
24 تُكتب DC Y (تعبير مطلق أو قابل للتقل)  
25 اصطفاك على نصف الكلمة . الطول الضمني نصف كلمة .  
26 الأطوال الظاهرة للمكثفة هي من 1 أو 2 بايتة .  
27 يتر إلى اليسار . يمكن التحديد في كود حرقي .  
29 لاحظوا أنّ التجهتين  
30 تساويان B و  $B + 2$   
31 الطول الظاهر  
32 يتر إلى اليسار  
35 ثوابت عنوان من النوع S  
36 تُكتب DC S (تعبير مطلق) .  
37 أو DC S (تعبير قابل للتقل) .  
38 أو DC S (تعبير مطلق) .

## السطر الملاحظة

الصفحة

73

- 39 موزونة في نصف كلمة . مصطفة حا . نصف الكلمة .
- 40 لا يمكن تحليها في كود حرفي .
- 42 القاعدة (Base) 0 ، الإزاحة (Déplacement) = 1024
- 43 قاعدة وإزاحة RELOC
- 49 ثوابت عنوان من النوع V
- 50 تستعمل فقط للمعايير الخارجية من النوع اسم البرنامج NOU-DE-PROG .
- 51 تكتب VDC (رمز خارجي قابل للتل) )
- 52 لا يرد الرمز للقابل للتل في أمر خارجي .
- 53 الطول الضمني 4 بابتات . ممثل الطول = 3 أو 4
- 54 اصطفا على حد كلمة ، بإمكانه أن يظهر في كود حرفي .
- 55 يولد للوزن كلمة صفر .

79

- 3 متالية الدخول
- 4 5 حفظ المرافف من شحن مرصف القاعدة
- 6 RI2 = مرصف القاعدة
- 7 البرنامج التالي
- 14 اصطفا كلمة
- 18 (1) القاعدة 12 ظاهرة
- 19 (2) القاعدة 12 ظاهرة
- 22، 23 كل الصليات من (3) حتى (7) تشحن "X" ABCDEF في المرصف 3 . الكلمة (3) هي الوحيدة
- المتعلقة عن مكان ALPHA بالإنسية إلى عنوان القاعدة .
- 24 و25 (3) استعمال تعبير قابل للتل . قاعدة ضمنية .
- 26 (4) تملجة تمثل رمزا مطلقا .
- 30 (7) استعمال كود حرفي
- 32 (8) "8" هي عبارة عن إزاحة
- 24 (9) خطأ اصطفا
- 36 (10) "12" هي عبارة عن إزاحة
- 37 (11) "12" هي عبارة عن مرصف قاعدي
- 38 (12) خطأ في النص
- 39 (13) خطأ في النص
- 40 (14) خطأ في النص
- 41 (15) 12 هي عبارة عن مؤشر

111

- 3 مؤشر (مضوب) إلى عنصر من TAB
- 4 مرصف إضافة لـ BXLE
- 5 مرصف مرجع لـ BXLE
- 6 مرصف عمل
- 11 القاعدة = المرصف 12
- 16 تصغير (إعداد)
- 23 طول الكلمة

111

- 47 حقل مسح الجدول  
48 و49 في حال عدم التبديل يتم فرز (ترتيب) TAB  
52 تصغير المؤشر  
53 تصغير مرجع EXLE  
56 عنصر يمر في مرصف العمل  
57 مقارنة  
60 تبديل  
62 تحديد موقع INDIC  
101 منطقة المخطات  
102 عدد عناصر TAB  
105 إحداد INDIC

115

- 3 مؤشر بداية الجدول التقوي  
40 مؤشر نهاية الجدول التقوي  
41 مؤشر للتصف والرتبة  
42 مرصف العمل  
43 طول المتصر  
62 عدد عمليات التكرار في البرنامج  
76 إحداد  
8 حساب عنوان المتصر الوسط (للتصف)  
84 قسمة  $L \div 2$   
85 (PTRELEM) = عدد العناصر في الجدول التقوي  
87 إذا 0 نزلج حتى 1  
89 ضرب بـ <  
91 مقارنة  
92 تفريع إذا كان  $ELEM < (MOT)$   
93 تفريع إذا كان  $ELEM > (MOT)$   
95 وجدنا المتصر حساب رتبة المتصر = (MOT)  
98 قسمة حل الطول  
100 طباعة للرتبة والقيمة

116

- 129 لم نجده  
153 منطقة للمطيات  
154 عدد كليات الجدول  
155 طول المتصر

160

- 3 حفظ مرصيف للتادي  
5 تعريف وشحن مرصيف النافذة  
6 تأخذ المرصيف I2 كنافذة  
7 عنوان PROG في I2  
8 حفظ R13 في المنطقة SAVE AREA من البرنامج .

## المصنعة

- 10 و 11 - حفظ عنوان المنطقة SAVE AREA من هذا البرنامج في المنطقة AREA من التالي 160  
16 تعريف المنطقة SAVE AREA  
22 متتالية نداء PROGK  
29 متتالية العودة إلى PROG1



## فهرست

الموضوع	الصفحة
تقديم .....	5
تمهيد .....	7

### القسم الأول : عموميات

1- الآلة البسيطة .....	9
2- تكوين المعلومات .....	20
تمارين .....	35
3- العنوان المطلقة ، العنوان النسبية .....	36
4- هيكلية الحاسبات IBM 360/370 .....	41
5- لغة الآلة .....	45
6- لغة المؤول .....	51

### القسم الثاني 360/370

7- العناصر الأساسية .....	59
8- توجيهات تعريف الرموز .....	67
تمارين .....	75
9- كتابة العناوين بلغة المؤول .....	76
10- التعليمات بلغة المؤول ، عموميات .....	81
11- الحساب بفاصلة ثابتة والحركات .....	84
تمارين .....	92
12- التصريحات .....	93
تمارين .....	98

99	13- العمليات المنطقية
104	تمارين
106	14- عمليات الإزاحة
109	تمارين
110	15- مسائل
117	16- الحساب العشري
120	17- الحساب بفاصلة متحركة
123	18- تعليمات التحويل والتمثيل
127	تمارين
129	19- الانقطاع والادخال والاخراج
138	20- الأوامر المتعلقة بالعنونة وتركيب المربيع
152	21- البرامج الثانوية
162	22- التأويل المشروط وماكرو التعليمات
177	23- نصائح في البرمجة
181	حلول التمارين
189	ملحقات
190	جدول تكويد السمات
192	جدول أبجدي للتعليمات
195	أوامر المؤول
196	مميزات الثوابت
197	الكود الحرفي موسّع







## هذا الكتاب

تعتبر لغة المؤول ( الأسمبر ) من العناصر الأساسية في التفكير حول طريقة البرمجة بإحدى اللغات المتطورة فهي تتيح لنا فهماً مفصلاً لأليات الحاسب وليس بالإمكان الاستغناء عنها في إعداد المعلوماتي .

وتتجلى ضرورة استعمال لغة المؤول ، بالرغم من قوة اللغات المتطورة ، عندما يوجد التزامات بالنسبة لفترات الإجابة ( بعض البرامج الكبيرة ، أنظمة التشغيل ، المصروفات ، الوقت الحقيقي ، ... ) أو بالنسبة لحجم الذاكرة ( الحاسبات الصغيرة والمتوسطة ) ، أو أيضاً للزامات تعود إلى عدم كفاية إمكانيات البرامج ( فورتران ، باسيك ) .

من جهة أخرى ، سوف يجد مستعملو الميكرومعلوماتية في تطبيق لغة المؤول حلاً ممتازاً لما يعترضهم من مشاكل .

يتوجه هذا الكتاب إلى الطلاب والممارسين الذين يرغبون بتعميق معرفتهم في مجال المعلوماتية . وهو يتكون من فصول قصيرة ويتبدى انطلاقاً من ملاحظات بسيطة جداً على حاسبة الجيب ، بشكل يقود معه القارئ شيئاً فشيئاً ، لا سيما بفضل التمارين المحلولة والمفاهيم الأساسية في بنية الآلة ، إلى دراسة المؤول والماكرو - لغة . ولا شك أنه بالإمكان استعماله كمرجع ولتدريس متعلق بسلسلة الآلات المعتمدة كأمثلة ( سلاسل 3000 ، 370 IBM ) ولكنه وضع كي يكون دليلاً عاماً يوجه بطريقة سليمة أي برمجة بلغة المؤول .